ISD97, A COMPUTER PROGRAM TO ANALYZE DATA FROM A SERIES OF *IN SITU* MEASUREMENTS ON A GRID AND IDENTIFY POTENTIAL LOCALIZED AREAS OF ELEVATED ACTIVITY

Marcel Reginatto, Peter Shebell and Kevin M. Miller

Environmental Measurements Laboratory
U.S. Department of Energy
201 Varick Street, 5th Floor
New York, NY 10014-4811

October 1997

19980903 072

DTIC QUALITY INSPECTED 1

DISCLAIMER

AQ I 98-11-2342

ABSTRACT

A computer program, ISD97, was developed to analyze data from a series of *in situ* measurements on a grid and identify potential localized areas of elevated activity. The ISD97 code operates using a two-step process. A deconvolution of the data is carried out using the maximum entropy method, and a map of activity on the ground that fits the data within experimental error is generated. This maximum entropy map is then analyzed to determine the locations and magnitudes of potential areas of elevated activity that are consistent with the data. New deconvolutions are then carried out for each potential area of elevated activity identified by the code. Properties of the algorithm are demonstrated using data from actual field measurements.

TABLE OF CONTENTS

INTRODUCTION

An important requirement in surveying for residual radioactivity is the detection of localized areas of elevated contamination, sometimes referred to as hot spots. We have developed a computer program that analyzes data from a series of *in situ* gamma-ray spectrometry measurements on a grid and searches for hot spots that might be "hidden" in the data. The program is designed to be used as a tool when evaluating compliance with regulations that set limits on the size and magnitude of hot spots. Since the program uses *in situ* measurements, the data are nuclide specific and reproducible. Therefore, our method is useful where standard techniques, such as scanning surveys, do not have the necessary sensitivity.

ANALYSIS OF *IN SITU* MEASUREMENTS ON A GRID

In the *in situ* gamma-ray spectrometry technique, a detector typically positioned 1 m above the ground facing downwards is used to measure the spectral distribution of photon fluence due to gamma emitting radionuclides in the soil. The gamma emitting radionuclides can be identified by their specific photon energies, which are registered as peaks in the spectrum. The peak count rate is related to the full absorption of unscattered gammas. If the detector is properly calibrated, the activity per unit mass for a given radionuclide of interest can be derived from the peak count rate using parameters that describe the soil characteristics (such as composition, density, and water content) and the depth profile of the distribution. The reader is referred to Beck et al. (1972), ICRU 53 (1994), Miller and Shebell (1995), and Huffert and Miller (1995) for more information on *in situ* gamma-ray spectrometry.

The *in situ* technique is particularly well suited for quickly determining levels of contamination over large areas. Each measurement provides a weighted average over the detector field of view, which is on the order of many square meters. For example, at a detector height of 1 m, about 30% of the total fluence rate measured comes from beyond a 3-m radius for ^{137}Cs that is uniformly distributed with depth in the soil. For a surface source distribution, about half the fluence comes from beyond 10 m (ICRU 1994). A series of measurements done on a grid on the order of meters apart will, therefore, provide overlapping coverage of the ground area.

A grid spacing close enough for overlapping coverage is required to ensure that any areas of elevated activity between grid points are not missed. For an elevated area to be detected, the minimum excess peak count rate at a detector due to the elevated area must exceed the uncertainty of the measurement. That is, the elevated area has to be of enough intensity to be seen above the level of noise. In this report, it will be assumed that the *in situ* measurements are made close enough so that there is overlapping coverage of the ground area. For most radionuclides of interest, a separation of 5 m is adequate to ensure overlapping coverage.

Although a series of measurements performed on a grid provides valuable information on the spatial distribution of the contaminant, a unique continuous solution cannot be derived from the set of measurements (this observation is not restricted to data collected with the *in situ* technique but holds generally for any set of measurements on a grid, such as soil samples). This is an important point that can be understood intuitively, since determining a continuous distribution from a finite number of measurements without making assumptions about the distribution is clearly not possible. Difficulties will also arise due to experimental errors associated with the set of measurements.

To select one distribution among those that satisfy the data requires additional information regarding the pattern of the deposition of the radioactivity on the ground. In some special cases where the process that causes the deposition of radioactivity on the ground is well understood, making some simplifying assumptions regarding the distribution is possible. One such case is undisturbed fallout from nuclear tests where the deposition is caused by the precipitation of radioactive material from a radioactive cloud of a large size. Here, one expects deposition that is roughly constant over areas that contribute to the detector's count rate (a circle of a few meters in diameter). Using this assumption, selecting a unique distribution of activity on the ground from among the many that fit the data is then possible. However, for many cases of interest, there are no reasonable assumptions that can be made regarding the pattern of activity on the ground. These include areas for which the history of contamination is not well known, and areas where activities such as remediation can affect the level of contamination. Here, the analysis of the data might require considering many possible scenarios that fit the data and checking to see if any such distributions will exceed regulatory limits. The code ISD97 was written to implement this approach. Given a set of *in situ* measurements on a grid, the code ISD97 determines locations and magnitudes for potential areas of elevated activity, and for each it generates a potential distributions of activity on the ground that fit the data. In practical applications, if any of these potential distributions appear to exceed the regulatory limits, further field work (such as soil

sampling, collimated measurements, additional measurements on a closer grid, etc.) would be used to determine whether such an elevated area was present.

Deconvolution of insitu Measurements on a Grid

The results from a set of measurements performed over a grid can be plotted on a map and contours drawn to show variability in the count rate and to delineate elevated areas (an example of such a contour plot is shown in Figure 1). It is often assumed that the variability of activity in the ground follows the pattern of the contour plot of the count rate. This is the simplest assumption that can be made, since one can easily calculate conversion factors from count rate to deposition on the ground (using the assumption of spatial uniformity or constant activity) and then "plug in" those conversion factors to obtain deposition on the ground. In general, the distribution of activity on the ground constructed in this way will not fit the measured data, since the assumption of spatial uniformity obviously does not hold. In addition, since contour plots are constructed by interpolation there is a tendency to generate smooth distributions in this way. Contour plots will not allow for any result that exceeds the highest value measured at a grid point, and will not single out potential elevated areas that might be present between the grid points.

To illustrate the difficulties that can be encountered if one attempts to interpret a contour plot as a map of deposition on the ground, consider the simplified case of three measurements on a line. Figure 2 shows three detectors far apart with little overlap in the fields of view, while Figure 3 shows three closely spaced detectors. The use of conversion factors to calculate deposition on the ground assumes a uniform distribution over the field of view of the detector, but this assumption cannot be satisfied where the field of view of different detectors with different count rates overlap. Between measurement points one has to interpolate between different values of activity (which correspond to the different count rates of neighboring detectors), and this introduces an error. In Figure 2 where there is little overlap, the error due to this interpolation will be small, and the contour plot will provide a map of smoothed activity on the ground that fits the data. However, this is a bad setup for a survey because good overlapping coverage of the ground area is essential to assure that elevated areas that fall "between" the grid points are not missed. If we insist on having complete coverage of the ground, we must consider the case illustrated in Figure 3. Here, since there is substantial overlap in the field of view of the detectors, the error due to interpolation will be large. In summary, when the grid of measurements is made

small enough to provide good coverage, a nonuniform distribution of activity on the ground that follows the contour plot will not fit the data.

To analyze data such as those provided in Figure 1 requires more than just preparing a contour plot. It requires carrying out a deconvolution of the data. In deconvolution, one uses the data together with a set of additional conditions to find a distribution of activity on the ground that fits the data and satisfies the given set of conditions. These additional conditions are needed because, as stated before, reconstructing the real distribution of activity on the ground from a finite number of measurements is an ill posed problem, and without additional requirements the solution is not unique. For a given distribution of activity on the ground $f(x)$, the peak count rate N_k measured at location k results from integrating the detector's response $R_k(x)$ over the $f(x)$, and is given by

$$N_k + \epsilon_k = \int R_k(x)f(x)dx \tag{1}$$

where ϵ_k is the (unknown) measurement error, the difference between the true value and the measured value. Deconvolution of the data amounts to finding a distribution $f(x)$ of activity on the ground that fits the data (Equation 1 for all measurements) and satisfies the given set of conditions. These conditions can be mathematical (e.g., one can choose the solution to be the smoothest function that satisfies the data) or these conditions may be derived from physical considerations (e.g., one can choose the solution that is closest to a given model while still satisfying the data). An important requirement in surveying for residual radioactivity is the detection of localized areas of elevated contamination, and one possible condition that can be imposed is to require that the solutions include the largest localized areas of elevated activity suggested by the data. This is of interest since regulations limit the magnitude and size of such areas, and given a set of data one would like to know if it is possible that a localized area of elevated activity might be "hidden" within that data. The computer code ISD97 unfolds the *in situ* data with the use of such a condition.

THE CODE ISD97

The computer code ISD97 is a FORTRAN 90 program that generates solutions to the deconvolution problem using a two-step process. The methodology is described in detail below, but for those already familiar with the maximum entropy method it can be summarized as follows. In the first part of the code, a deconvolution of the data is carried out using the maximum entropy principle with the assumption of a constant default (background) level. The output of the maximum entropy deconvolution is an activity distribution that tends to be flat almost everywhere except in the neighborhood of each measurement point, where there will be a localized peak or a depression depending on how the measured peak count rate at that measurement point compares with the average peak count rate. This makes it possible to associate with each measurement a given level of activity in the ground that can be used to identify the location and magnitudes of potential areas of elevated activity. In the second part of the code, the maximum entropy solution is analyzed, and this information is used to find the locations and magnitudes of potential areas of elevated activity. New deconvolutions are then carried out for each potential area of elevated activity. This second series of deconvolutions are done using the maximum entropy principle with default levels set equal to a constant (background) level, plus an area of elevated activity superimposed on it. A diagram summarizing the algorithm is shown in Figure 4, and a complete listing of the code is in Appendix B.

To formulate the deconvolution problem in discrete terms, we divide the distribution of activity on the ground into a large number of equal-sized cells, the ith cell having activity f_i. In this way, the continuous distribution of activity on the ground $f(x)$ is approximated by point sources f_i at each cell i. Throughout this report we use the following notation. The peak count rate measured at point k is N_k and σ_k is the statistical counting error at 1σ. R_{ki} is the matrix representing the instrument response (the count rate measured at point k per unit activity in cell i) and $N_k = \Sigma R_{ki} f_i$ is the calculated peak count rate at the measurement point k for a given activity distribution f_i. We use the notation $f_i^{\ def}$ to denote the default level of the cell i (default levels have to be specified for the maximum entropy deconvolution of the data). Finally, we use the expression "fit the data" to mean that the chi-square,

$$\chi^2 = \sum \epsilon_k^2 / \sigma_k^2 \tag{2}$$

is on the order of the number of measurements (and, therefore, the distribution of activity folded with the detector response will have for each measurement an error equal on average to the uncertainty of the measurements).

The optimal cell size is determined from the following considerations: the cells must be small enough to ensure that such an approximation does not introduce a substantial error in the calculation, but large enough to reduce the total number of cells needed to model the ground and in this way reduce the computational burden. Square cells with sides measuring 50 cm have proven adequate to balance these two conflicting needs. The potential areas of elevated activity are modeled as squares made from N^2 cells (where N is an integer, N=1,2, ...). Their size is chosen by the user. This choice will depend, in general, on the regulations that are applicable to the survey area.

All the information needed for the deconvolution of the data is read in at the beginning, either by the main program or by the subroutine ISD. The code requires an input file that defines:

(a) the number of cells used to model the ground, which is modeled as a rectangle, and the size of each cell (in cm),

(b) the number of measurements, and for each measurement k its coordinates x_k, y_k (in cm),

(c) the peak count rate N_k and associated uncertainty σ_k (in counts min^{-1}) for each measurement,

(d) the photon energy being measured (in keV),

(e) the detector ID code (the detector response is calculated by the subroutine RESPONSE, which includes the detector parameters), and

(f) an internal parameter used to establish a length scale for the problem (used in the subroutine PEAKXELS).

In addition, the user is prompted to enter the names of the input and output files, and the size of the potential areas of elevated activity. Finally, the user is given a choice for modeling the depth profile of the activity on the ground (uniform distribution or surface distribution), a choice of count rates to be used to set the background activity, and a choice of units to be used to report the final activity.

Since the potential areas of elevated activity have to be superimposed over a background activity, a choice of background activity needs to be made. This background activity can be estimated from either the data or it can be entered if known *a priori*. The uniform background distribution is calculated from the user's choice of one of five count rate values:

(a) the average count from all measurements,

(b) the median value of the count rates,

(c) the count rate from the lowest measurement,

(d) a weighted average that minimizes the chi-square of the background distribution, and

(e) a user determined count rate.

Choice (e) is recommended if the count rate due to the background activity can be estimated *a priori*, for example, from a neighboring reference area, while (d) is recommended if no information regarding the background distribution is available. The other choices made available to the user can be useful under special circumstances. Selecting (d) sets the background activity for each cell equal to

$$f_i \equiv \left(\sum_k N_k (\Sigma R_{ki}) / \sigma_k^2 \right) / \left(\sum_k (\Sigma R_{ki})^2 / \sigma_k^2 \right). \tag{3}$$

After the initial distribution is chosen, the next step is to carry out the maximum entropy deconvolution of the data. This is done by the subroutine MAXENT, with the f_i^{def} set equal to the background level. This completes the first part of the algorithm. The results of this first part is the determination of a background activity level (which normally will not fit the data) and a maximum entropy deconvolution of the data that fits the data. As we pointed out before, the maximum entropy solution tends to be flat almost everywhere except in the neighborhood of each measurement point, where there will be a localized peak or a depression depending on how the measured peak count rate at that measurement point compares with the average peak count rate.

In the second part of the code, the maximum entropy solution is analyzed to determine the positions and intensities of potential localized areas of elevated activity that might be present in

the data. The subroutine PEAKXELS associates an integrated activity (IA) with each measurement by adding the activity that lies within a circle centered about the measurement point and subtracting from it the background activity. The circles are made as large as possible without having them overlap. The IAs will be positive or negative depending on whether there is an excess or a deficit of activity associated with that measurement point. We point out that making a quantitative correspondence between measurement points and localized activity using the peak count rates is not trivial due to the overlapping fields of view of the detectors and the nonuniformity of their response functions (which close to the measurement point varies roughly as the inverse distance squared). The assumption is made that any potential areas of elevated activity will be found in the neighborhood of those measurement points where the IA is positive. For each potential area of elevated activity, the code carries out a full analysis (as described below) that includes in each case a new deconvolution of the data.

After determining that a measurement point has a positive IA, the subroutine FINDHS is used to find the location of the area of elevated activity associated with that measurement point. This is achieved by comparing the IA at that measurement point with the IAs of the neighboring measurement points (for example, for a square grid the code compares the IA of the measurement point in question with those of the eight measurement points closest to it). FINDHS uses the inverse distance squared relationship and finds the optimal location for a point source by choosing a location that minimizes the error incurred when the activity in the IAs of neighboring measuring points is replaced by the point source (the procedure is described in more detail in the section on the subroutine FINDHS). After the positions of the potential areas of elevated activity are fixed, the magnitudes are determined by the subroutines MAGPS and MAGPSNN. The magnitude chosen is the largest one that minimizes either the chi-square of the measurements in the immediate neighborhood of the point source, or the chi-square of the measurement that is closest to the area of elevated activity. Finally, new deconvolutions are done taking into account the information on the position and magnitudes of the potential areas of elevated activity. For these final deconvolutions, the f_i^{def} are set equal to the background level plus the area of elevated activity superimposed on it. Except for the way in which f_i^{def} are chosen, the deconvolutions proceed using the maximum entropy method as described before.

The output consists of four files, one file that includes tables of results and three files that contain maps of activity distributions for plotting. The first file, which has extension "TBL," includes:

(a) the name of the file,

(b) the detector ID code and photon energy,

(c) the average count rate and the value of the uniform activity associated with that average count rate,

(d) the background count rate and the value of the uniform activity associated with that background count rate,

(e) the chi-square of the uniform background activity distribution,

(f) the chi-square of the maximum entropy solution,

(g) parameters related to the largest potential elevated area, such as the location, magnitude, area, and the chi-square of the distribution associated with this elevated area,

(h) a table with the measured counts, calculated counts and the ratio of measured to calculated peak count rates at each measurement point for the activity distribution of the largest potential area of elevated activity, and

(i) a table with the coordinates and magnitudes of all potential areas of elevated activity, and the chi-square of the distribution associated with each elevated area.

The second file, which has extension "MXE," is the output from the maximum entropy deconvolution. The third file (which has a blank extension) is the activity distribution corresponding to the largest potential area of elevated activity over a smooth background. The fourth file, which has extension "ALL," is a collection of all the potential areas of elevated activity plotted over a smooth background.

ALGORITHMS

A description of the algorithms used in the most important subroutines follows.

THE SUBROUTINE MAXENT

We now present the maximum entropy deconvolution algorithm used in the code, which is a modification of the one in Wilczek and Drapatz (1985). For a discussion of the maximum entropy principle, we refer the reader to Appendix A, and to the references contained therein.

Using the notation described at the previous section, we define the set of distributions that fit the data using two restrictions:

$$N_k + \epsilon_k = \sum R_{ki} f_i, \tag{4}$$

$$\sum \epsilon_k^2 / \sigma_k^2 = \Omega, \tag{5}$$

where Ω is a parameter that equals the value of the χ^2 of the solution (typically, Ω is set equal to the number of measurements), k labels the measurement (k=1, ..., m) and i labels the cell (i=1, ..., n). The set of equations (4) is the discretized version of (1). Equation (5) is a χ^2-constraint for handling the (unknown) errors ϵ_k and assumes that the errors are normally distributed with zero mean and variances σ_k^2. From this set of admissible distributions we want to select the one that maximizes the entropy S of the distribution,

$$S = - \sum \{f_i \log (f_i / f_i^{def}) + f_i^{def} - f_i\} \tag{6}$$

where the f_i^{def} is the (discretized) default distribution. Equation (6) is in the form given in Skilling (1989).

The Lagrangian associated with the maximization of (6) with constraints (4) and (5) is of the form

$$L(f_i, \epsilon_k, \lambda_k, \mu) = -\sum \{f_i \log(f_i / f_i^{\,def}) + f_i^{\,def} - f_i\} \tag{7}$$
$$-\sum \lambda_k \{\sum R_{ki} f_i - N_k - \epsilon_k\} - \mu \{\sum(\epsilon_k / \sigma_k)^2 - \Omega\}.$$

where λ_k, μ are (m+1) Lagrange multipliers. Variation with respect to f_i, e_k and μ lead to the set of (n+m+1) equations

$$-\log(f_i / f_i^{\,def}) - \sum \lambda_k R_{ki} = 0 \tag{8}$$

$$\lambda_k - 2\mu\epsilon_k/\sigma_k^2 = 0 \tag{9}$$

$$\sum \epsilon_k^2/\sigma_k^2 = \Omega. \tag{10}$$

We can use (8), (9) and (10) to solve for the f_i, the ϵ_k and μ in terms of the λ_k,

$$f_i = f_i^{\,def} \exp\{\sum\lambda_k R_{ki}\} \tag{11}$$

$$\epsilon_k = \lambda_k\sigma_k^2/2\mu \tag{12}$$

$$\mu = (\sum(\lambda_k\sigma_k)^2/4\Omega)^{1/2}. \tag{13}$$

Variation with respect to the λ_k leads to (4), and using (11), (12), and (13) we can write these m equations as

$$N_k + \lambda_k\sigma_k^2 (\Omega/\sum(\lambda_j\sigma_j)^2)^{1/2} - \sum R_{kj} f_j^{\,def} \exp\{\sum\lambda_l R_{lj}\} = 0 \tag{14}$$

Thus, the initial optimization problem stated has been reduced to a system of m equations with m unknowns λ_1, ..., λ_m. These equations can also be derived by maximizing a potential function Z,

$$Z = -\sum f_i^{\text{def}} \exp\{-\sum \lambda_k R_{ki}\} - (\Omega \sum (\lambda_k \sigma_k)^2)^{1/2} - \sum N_k \lambda_k \qquad (15)$$

with respect to the λ_k. Therefore, we can reformulate the problem in terms of the maximization of the potential function Z.

The f_i^{def} are default levels for each cell i . Without the chi-square constraint, the maximum entropy solution would default to $f_i = f_i^{\text{def}}$. When the subroutine MAXENT is called the first time, the f_i^{def} are set equal to a constant, background activity distribution. When it is called for the final deconvolutions (after the initial magnitude of each of the potential areas of elevated activities are determined by the subroutines MAGPS and MAGPSNN), the f_i^{def} are set equal to a background level plus an area of elevated activity superimposed on it.

To find the extremum of the potential function Z, we use the subroutines FRPRMN and LMBDAS. FRPRMN is more robust while LMBDAS converges more quickly. For the deconvolution that gives us the maximum entropy map, we first use FRPRMN with the initial value of the λ_k set equal to zero, and then restart the maximization using LMBDAS. For all other deconvolutions, we use LMBDAS with the initial value of the λ_k set equal to the values that correspond to the maximum entropy solution. If LMBDAS stops converging the deconvolution proceeds using FRPRMN.

THE SUBROUTINE LMBDAS

This subroutine uses an iterative procedure based on the integral equations (14) from which follow the set of equations

$$\lambda_k^{M+1} = (1-p)\,\lambda_k^M + p\,(\sum R_{kj} f_j^{\text{def}} \exp\{\sum \lambda_l^M R_{lj}\} - N_k)\,(\sum (\lambda_j^M \sigma_j)^2/\Omega)^{1/2}/\sigma_k^2 \qquad (16)$$

where M denotes the Mth iterate. The parameter p is a smoothing parameter introduced to help convergence. It is set initially to 0.01 and allowed to increase to up to 0.1 in successive iterations. This iterative procedure requires initial values λ_j^0 satisfying

$$\sum(\lambda_j^0 \sigma_j)^2 \neq 0. \tag{17}$$

THE SUBROUTINE FRPRMN

This subroutine is taken from Press et al. (1992). Given a set of initial values for the λ_k, it uses the conjugate gradient method in multidimensions to maximize the potential function Z.

THE SUBROUTINE PEAKZELS

This subroutine calculates the excess or deficit of activity associated with each measurement position. The IA associated with each measurement is calculated by adding the activity that lies within a circle centered about the measurement point and subtracting from it the background activity f_i^{BKG},

$$IA_k = \sum \{f_i - f_i^{BKG}\} \tag{18}$$

where the sum is over the f_i within the circle, centered about the measurement position k. The circles are made as large as possible without having them overlap. The IAs will be positive or negative depending on whether there is an excess or a deficit of activity associated with that measurement point.

THE SUBROUTINE FINDHS

This subroutine calculates the optimal position of the potential area of elevated activity. To find this position, we use the following reasoning. If measurements at neighboring measurement points are in fact due to a single localized area of elevated activity, then it should be possible to replace the IAs of these measurement points with excess activity at a single location and still have a distribution that fits the data.

To carry out this procedure, the potential localized area of elevated activity is approximated by a point source. For each neighboring measurement point k, and for a point source of activity A_e located at an arbitrary point e, we now define an error term δ_{ek} using the equation

$$\sum (IA_j/r_{jk}^2) = (A_e + \delta_{ek})/r_{ek}^2 \tag{19}$$

where IA_j is the integrated activity of measurement point j, r_{ek} is the distance between the cells e and k, and the sum is over the measurement points close to the area of elevated activity. Here, we have taken into account the fact that the detector response for a point source can be approximated by the inverse distance squared relationship. For a given location e, the activity A_e is found by minimizing the magnitude of the error term vector E_e associated with the point e, which we define as

$$E_e = (\sum \delta_{ek}^2)^{1/2}. \tag{20}$$

By systematically calculating E_e for different locations e close to k, one can determine the location at which it is a minimum, and in this way define the optimal position of the potential area of elevated activity. While this procedure involves introducing approximations and it is, therefore, not exact, it has been found to be reliable and to lead to good results.

THE SUBROUTINE MAGPS

This subroutine calculates the optimal intensity of the potential area of elevated activity. The area of elevated activity is modeled with a point source at cell e superimposed on a flat background, and the activity of the point source is calculated by minimizing the chi-square of the measurements in the immediate neighborhood of the point source.

THE SUBROUTINE MAGPSNN

This subroutine provides an alternative calculation of the optimal intensity of the point source used in the modeling of a potential area of elevated activity. The area of elevated activity is modeled with a point source at cell e superimposed on a flat background, and the activity of the

point source is calculated by minimizing the chi-square for the measurement that is nearest to the point source only. The value of the activity calculated by MAGPSNN is used only when it exceeds the activity calculated by MAGPS.

THE SUBROUTINE RESPONSE

This subroutine contains all the relevant information on the instrument response (the count rate per unit activity) for the detectors used, and calculates the response matrix R_{ki}. Currently, the calculation of the response matrix takes into account the inverse distance squared relationship, the photon energy, the photon attenuation in air and ground, detector parameters related to the efficiency and angular response, and a choice of either a uniform or a surface source distribution to model the distribution in depth. This subroutine is in a separate file, and can be easily modified and updated.

TEST CASES

The computer code ISD97 has been tested using both simulated data and actual measurements. In this section we summarize results obtained with data from two different field measurements.

TEST CASE NUMBER ONE

We summarize the results of an analysis of a series of field measurements where a 28.23 kBq ^{137}Cs check source was used to simulate an area of elevated activity. The results described here differ from the analysis in Reginatto et al. (1996) due to a number of improvements that have since been incorporated into the code. In particular, the maximum entropy algorithm and the subroutines that calculate the position and magnitude of the potential areas of elevated activity have been rewritten, resulting in better agreement between the parameters calculated by the code and the actual position and magnitude of the check source used to simulate the area of elevated activity.

The field site chosen was nonuniform, including some areas that had been resurfaced. The ^{137}Cs background distribution varied between close to zero and its expected global fallout level in this area (2.96 kBq m^{-2}). Measurements were made with a 40% relative efficiency germanium

detector placed at a height of 1 m, with a counting time of 10 min at each measurement point. Thirty-five equally spaced measurements of the 662 keV γ fluence were taken on a rectangular 7×5 grid. The measurement points were 2.44 m (8 ft) apart. The statistical uncertainty was < 5% (1σ) for each measurement. The peak count rate at different measurement points varied by up to a factor of eight due to the variability of the background and to some extent also to the presence of the area of elevated activity. The data was processed in two different ways. In one case, all 35 measurements were taken into account. In the other, every other measurement was discarded, resulting in a total of 12 measurements on a 4x3 rectangular grid with a spacing of 4.88 m (16 ft). For the deconvolution, the area of elevated activity was modeled as one cell, a square 0.4 m × 0.4 m in size.

Figure 1 shows the peak count rate measured at each measurement point in the grid, along with a contour plot constructed from the data by interpolation. The position of the check source is also shown. It is common practice to assume that the activity on the ground follows a similar contour pattern. As pointed out before, this approximation is not valid when there is good overlap of the field of view of the detectors. For our test case, this is certainly not so, and the contour plot does not provide a good map of the underlying activity distribution that will fit the data. Figure 5 shows the output from the maximum entropy deconvolution, while Figure 6 is the activity distribution that corresponds to the largest single potential area of elevated activity over a smooth background for the 35 measurements case. Both types of deconvolutions generate good solutions in the sense that in both cases the activity distributions fit the data. The maximum entropy solution shows structure that reflects the grid geometry with peaks centered exactly at measurement points with an excess of activity. The solution with the largest area of elevated activity on the other hand has one large peak that falls between measurement points superimposed on a smooth background.

The code located the potential area of elevated activity very precisely, within 22 cm of the actual location of the check source when using 35 measurements, and within 37 cm when using 12 measurements (the code also provided locations for other potential areas of elevated activity that turned out to be related to areas that had higher levels of activity due to variations in background activity, but we will restrict this discussion to the potential area of elevated activity that corresponds to the check source). The code was also very successful in estimating the magnitude of the point source. The calculated magnitude of the point source was 26.93 kBq (within 5% of the true value) when using 35 measurements, and 75.84 kBq (2.7 times the true value) when using 12 measurements. The 12 measurements case is less constrained, and, therefore, allows for a larger potential area of elevated activity.

TEST CASE NUMBER TWO

We summarize an analysis of *in-situ* measurements in a room that had ^{137}Cs contamination on the floor. The floor of the room was rectangular and measured 5.0 m × 6.0 m. Four measurements at 662 keV γ fluence with a counting time of 10 min were taken with a 40% relative efficiency germanium detector placed at a height of 1 m. To set up the grid, the floor was divided into four quadrants with a measurement centered in each. The statistical uncertainty was on the order of 5% for each measurement, and the peak count rate at different measurement points varied by up to a factor of three due to the presence of the contaminant. In addition to these measurements, 22 collimated measurements taken with the detector placed close to the ground and using a collimator that restricted the field of view to a circle of 35 cm in radius. These collimated measurements were not analyzed with the code ISD97, but were used instead to construct a map of localized activity that can be compared with the results of the deconvolution of the four uncollimated measurements. For the deconvolution, the area of elevated activity was modeled as a square 1.0 m × 1.0 m in size. The background level (which was essentially zero) was set equal to the statistical uncertainty. Further details can be found in Miller et al. (1997).

Figure 7 shows a map of the floor with peak count rates plotted for both uncollimated and collimated measurements, and the location of the four uncollimated measurements. In addition, the location of the potential elevated area identified by the code is shown. Although the contamination appears to extend over an irregular area of more than one square meter, the location of the area of elevated activity identified by the code corresponds with good precision to the contaminated area. The code was also very successful in estimating the magnitude of the elevated area. The activity calculated by the code from the four uncollimated measurements equals 0.41 Bq, while the activity derived from the four highest collimated measurements in the neighborhood of the elevated area equals 0.48 Bq (the total inventory calculated from all measurements equals 0.52 Bq).

The data from a few uncollimated measurements, when analyzed with the code ISD97, can provide information that would otherwise require many collimated measurements. Use of this methodology can, therefore, result in substantial savings in time and expense.

CONCLUSIONS

In this report we describe in detail the code ISD97, a computer program that analyzes data from a series of *in situ* measurements on a grid and identifies potential areas of elevated activity. Due to the nonuniqueness of the deconvolution problem, we do not search for a single solution that fits the data, but have developed instead a methodology in which we search for many distributions of activity on the ground, all of which fit the data and have a potential area of elevated activity. This approach seems to be the only one adequate for areas for which the history of contamination is not well known, as well as for areas where activities such as remediation can affect contamination in unforeseeable ways, since then there are no reasonable assumptions that will select one solution from all the possible solutions that fit the data.

The methodology presented here provides an alternative to the standard methods of detecting areas of elevated activity. Of the methods currently in use, one of the most common is a "walkover" using a survey meter gross count rate to signal the presence of an area of high radioactivity. These surveys depend to some extent on the expertise of the technician doing the "walkover" and (unlike regulations) do not discriminate between different radionuclides. The results of such surveys are, therefore, more qualitative than quantitative. The method implemented in the code ISD97 uses *in situ* data taken under controlled conditions. It provides the location of potential areas of elevated activity consistent with the data, and it lists activity levels for all the potential areas of elevated activity found. In this way a quantitative record of the magnitude of areas of elevated activity that might be "hidden" in the data is generated. Analyses of data from actual field measurements (two of which are summarized in this report) show that the methodology is directly applicable to practical cases.

REFERENCES

Beck, H. L., J. De Campo and C. K. Gogolak
In Situ Ge(Li) and NaI(Tl) Gamma-Ray Spectrometry
U.S. Atomic Energy Commission, New York, Report HASL-258 (1972)

Huffert, A. M. and K. M. Miller
Measurement Methods for Radiological Surveys in Support of New Decommissioning Criteria
NUREG-1506 (Draft Report for Comment),Washington, D.C.: Nuclear Regulatory Commission
  (1995)

International Commission on Radiation Units and Measurements
Gamma-Ray Spectrometry in the Environment
Bethesda, MD: ICRU Publication 53 (1994)

Jaynes, E. T.
Information Theory and Statistical Mechanics
Phys. Rev. 106:620-630 (1957)

Jaynes, E. T.
Prior Probabilities
IEEE Trans. Syst. Cybern. SSC-4:227-241 (1968)

Miller, K. M. and P. Shebell
In Situ Gamma-Ray Spectrometry - A Tutorial for Environmental Radiation Scientists
USDOE Report EML-557 (1995)

Miller, K. M., M. Reginatto, P. Shebell, G. A. Klemic, and C. V. Gogolak
Spectrometric Techniques for the Measurement of Radioactive Contamination in Buildings
Proceedings Air and Waste Management Association's 90th Annual Meeting and Exhibition
  Toronto, Ontario, Canada, June 8-13 (1997)

Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery
Numerical Recipes in FORTRAN: the Art of Scientific Computing
Cambridge University Press, New York, NY, Second Edition (1992)

Reginatto, M., P. Shebell and K. M. Miller
An Application of the Maximum Entropy Method for Assessments of Residual Radioactivity
 at Contaminated Sites
IEEE Trans. Nucl. Sci. 43:1837-1841 (1996)

Shannon, C. E.
A Mathematical Theory of Communication
Bell Syst. Tech. J. 27:379-423, 623-656 (1948)

Shore, J. E. and R. W. Johnson
Axiomatic Derivation of the Principle of Maximum Entropy and the Principle
 of Minimum Cross-Entropy
IEEE Trans. Info. Theory IT-26: 26-39 (1980)

Skilling, J.
Classic Maximum Entropy
In: J. Skilling (Editor), Maximum Entropy and Bayesian Methods
Kluwer Academic Publishers, Dordrecht (1989)

Wilczek, R. and S. Drapatz
A High Accuracy Algorithm for Maximum Entropy Image Restoration
 in the Case of Small Data Sets
Astron. Astrophys. 142: 9-12 (1985)
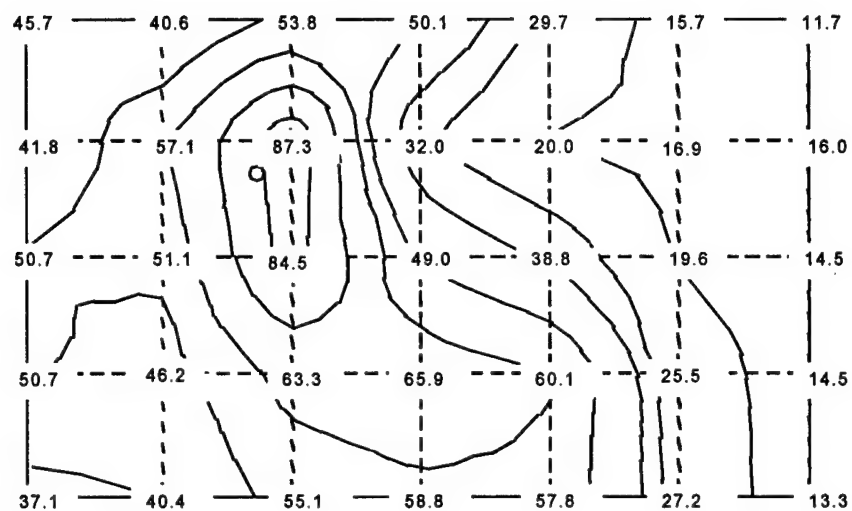
Figure 1. Grid with peak count rates (counts min^{-1}) at each measurement point. The circle indicates the placement of a check source used to simulate an elevated area.
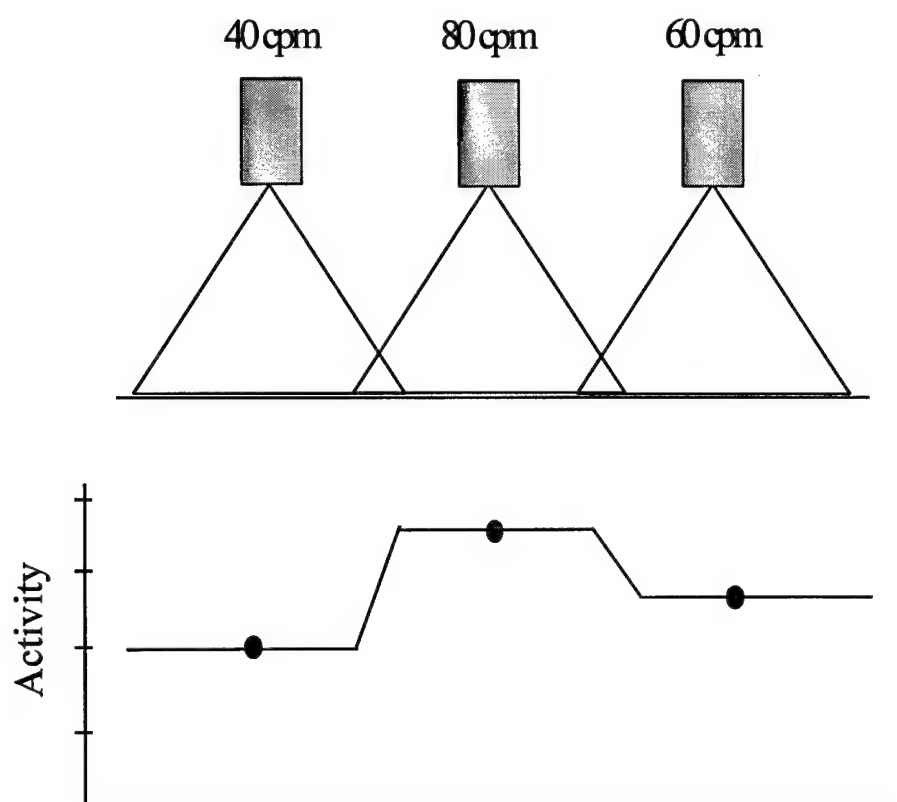
Figure 2. Map of activity on the ground (constructed from the count rate contour plot) for three detectors with little overlap.
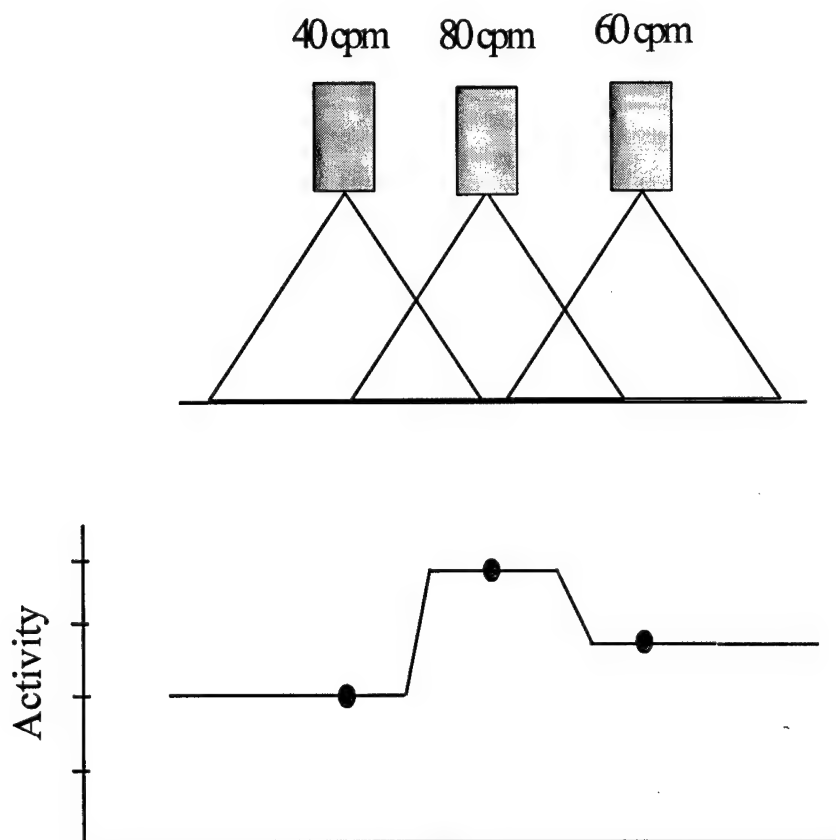
Figure 3. Map of activity on the ground (constructed from the count rate contour plot) for three closely spaced detectors.

READ INPUT
⇓
CALCULATE DETECTOR
RESPONSE MATRIX
⇓
CALCULATE ACTIVITY OF UNIFORM
BACKGROUND DISTRIBUTION
⇓
CALCULATE MAXIMUM ENTROPY SOLUTION
(MAXIMUM ENTROPY DECONVOLUTION WITH
DEFAULT DISTRIBUTION = BACKGROUND)
⇓
FOR ALL MEASUREMENT POINTS
CALCULATE INTEGRATED ACTIVITY (IA)
FROM THE MAXIMUM ENTROPY SOLUTION
⇓
FOR MEASUREMENT POINTS WITH IA > 0
CALCULATE POSITION OF
POTENTIAL ELEVATED AREA
⇓
FOR MEASUREMENT POINTS WITH IA > 0
CALCULATE MAGNITUDE OF
POTENTIAL ELEVATED AREA
⇓
FOR MEASUREMENT POINTS WITH IA > 0
PLACE POTENTIAL ELEVATED
AREA OVER BACKGROUND
⇓
FOR MEASUREMENT POINTS WITH IA > 0
CALCULATE ACTIVITY DISTRIBUTION THAT FITS DATA
AND CONTAINS A POTENTIAL ELEVATED AREA
(MAXIMUM ENTROPY DECONVOLUTION WITH DEFAULT
DISTRIBUTION = BACKGROUND + POTENTIAL ELEVATED AREA)
⇓
IDENTIFY LARGEST POTENTIAL
ELEVATED AREA
⇓
WRITE OUTPUT

Figure 4. Algorithm of code ISD97.
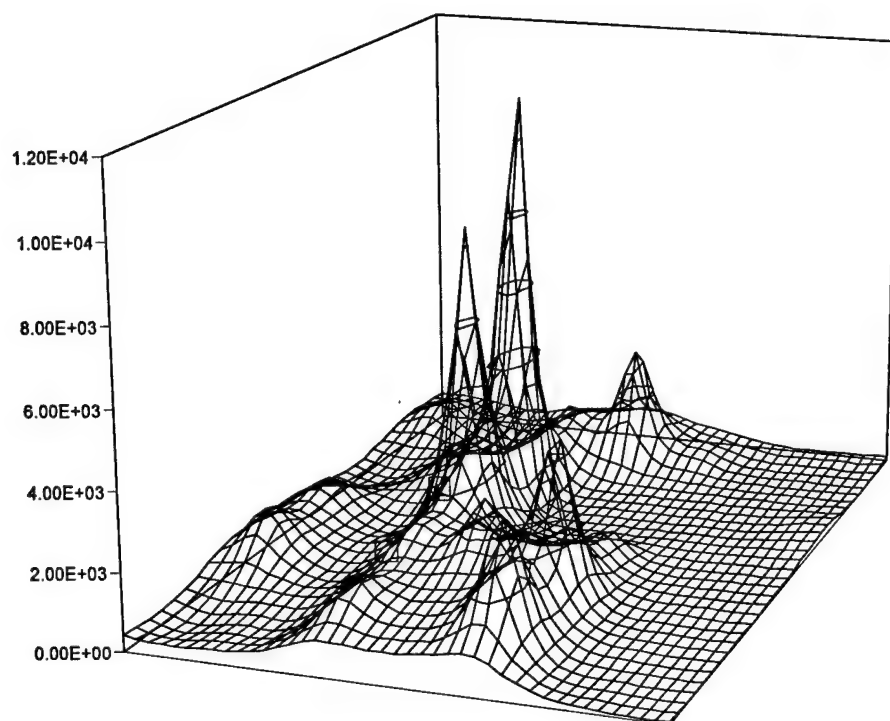
Figure 5. Activity distribution (Bq m^{-2}) from maximum entropy deconvolution (test case 1).
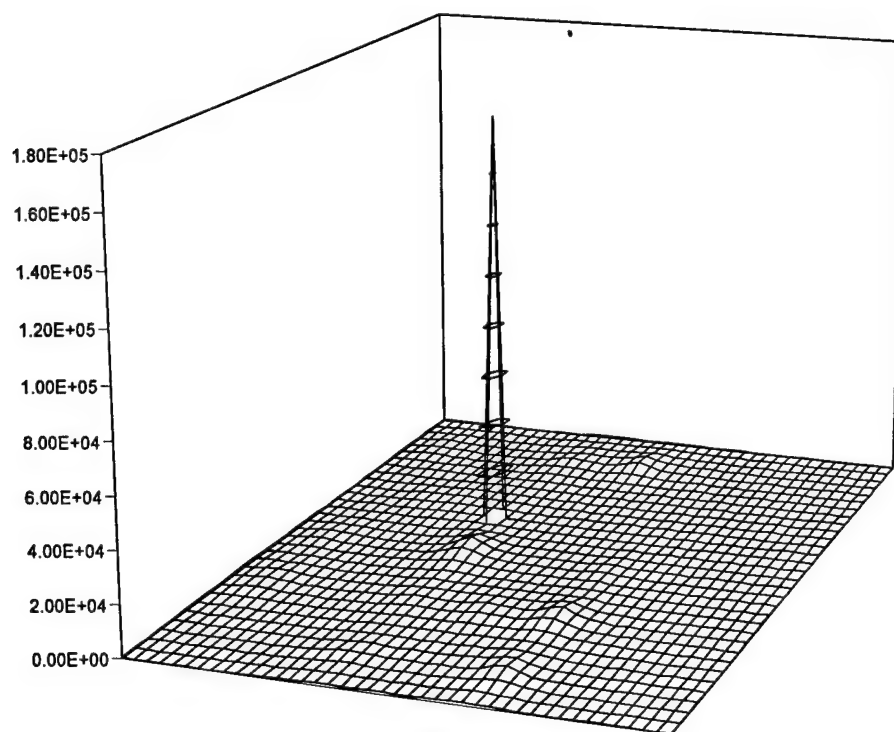
Figure 6. Activity distribution (Bq m^{-2}) for largest potential area of elevated activity (test case 1).

23.7   108.4   4.8
24.1   73.3   0.7
1m
1.9   121.4   3.3
3.7   43.3   0.0
0.7   NM   2.2
2.2   29.2   0.0
0.4   8.1   0.0
0.0   NM   0.0

Location of potential area of elevated activity

Location of uncollimated measurement

Location of collimated measurement and measured surface activity (kBq m^{-2})
(NM indicates that no measurement was made at that location)
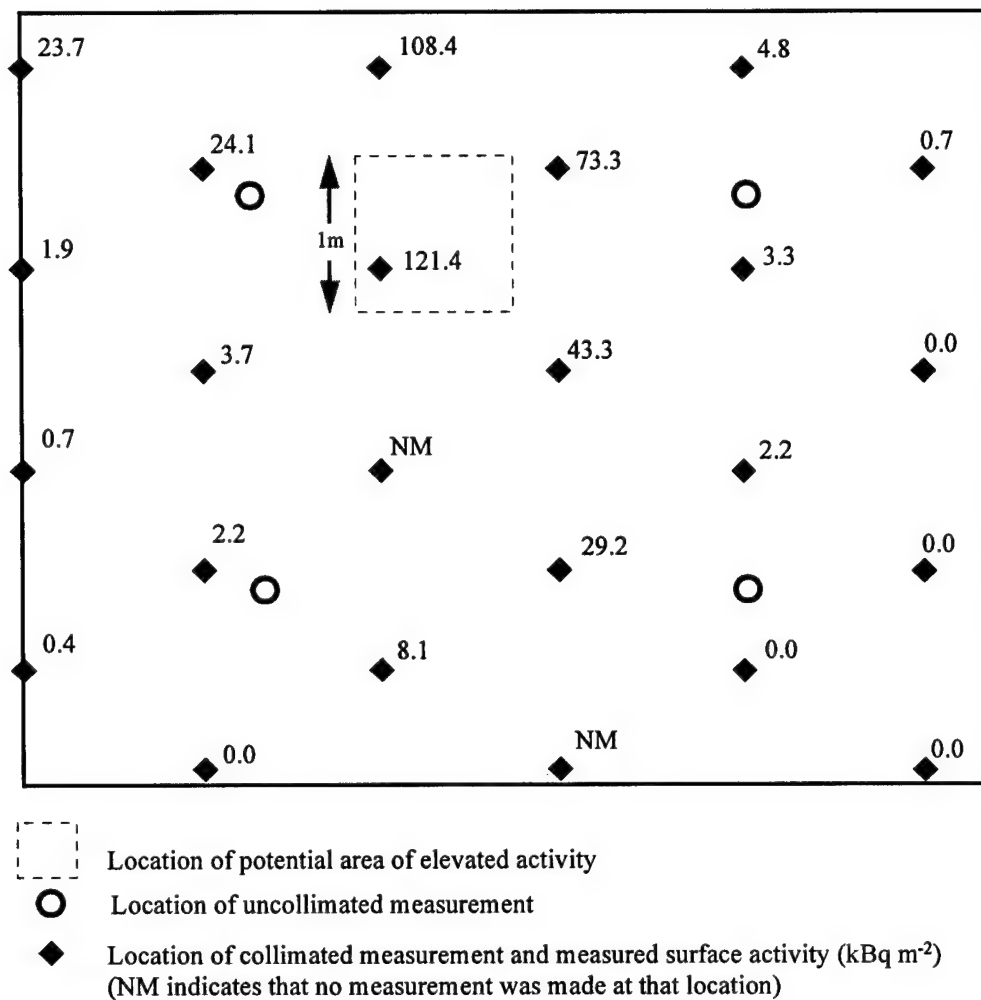
Figure 7.   Analysis of *in situ* measurements in a 5 m x 6 m room with ^{137}Cs contamination on the floor (test case 2). Data from collimated measurements, location of uncollimated measurements, and location of potential area of elevated activity identified by the code.

APPENDIX A

THE PRINCIPLE OF MAXIMUM ENTROPY

The entropy of the distribution f(x), in the form given in Skilling (1989), is given by

$$S = - \int \{ f(x) \log (f(x)/f(x)^{def}) + f(x)^{def} - f(x) \} \, dx, \qquad (A.1)$$

where $f(x)^{Def}$ is the default distribution. This is a generalization of the usual cross-entropy formula

$$S_{CE} = - \int \{ f(x) \log (f(x)/f(x)^{def}) \} \, dx, \qquad (A.2)$$

which appears as the first term in (A.1). The second term in (A.1) is an additive constant that ensures that the global maximum of S (at $f(x) = f(x)^{def}$) is zero, a sometimes convenient but not essential requirement. The third term in (A.1) ensures that in the absence of any other constraints S is maximized at $f(x) = f(x)^{def}$ (rather than just being proportional).

Before we go on, a note on terminology: some authors use the term entropy when referring to what we call here cross-entropy, while others use the term cross-entropy for the negative of S_{CE} and as a result end up with the principle of *minimum* cross-entropy. Other names for S_{CE} (or its negative) are relative entropy, direct divergence, expected weight of evidence, Kullback-Leibler distance, and Kullback number. To add to the confusion (and to keep the arguments as close to the standard literature as possible), when discussing the maximum entropy principle in this section, we will be referring to maximization of S_{CE}, not of S. For an axiomatic derivation of S (rather than S_{CE}), we refer the reader to Skilling (1989).

Historically, the concept of entropy originated in thermodynamics, but its generalization plays an important role in information theory, probability theory, and statistical inference. The maximum entropy principle provides a rule for determining a positive, additive distribution given definite but incomplete constraints. The probability distribution pr(x) of a variable x is an example of a positive, additive distribution. It is positive by construction, and it is additive in the sense

that the overall probability in a domain equals the sum of the probabilities in any decomposition into subdomains. In our case, the positive, additive distribution to be determined is the distribution of activity f(x), and the constraints are the measurements and the experimental errors associated with them. The maximum entropy principle states that one should choose from all of the distributions f(x) that satisfy the set of constraints the one for which S_{CE} is maximal. A parallel procedure is used in statistical mechanics, where given constraints on the total energy and number of particles of a gas, entropy maximization is used to infer a distribution that is uniform in space and Maxwellian in velocity. There are several approaches that lead to the maximum entropy principle, and we summarize two of them below.

In the original derivation by Jaynes (1957, 1968), the use of the maximum entropy principle is justified on the basis of the cross-entropy's unique properties as an uncertainty measure. Arguments originating in information theory show that the Shannon entropy,

$$S_{SH} = - \sum f_i \log (f_i / f_i^{\ def})$$ (A.3)

is the appropriate measure of the *amount of uncertainty expressed in a distribution.* These arguments follow from requiring three consistency conditions that are generally accepted as requirements for an uncertainty measure. Expression (A.3) for the case $f_i^{\ def} = 1$ was originally derived by Shannon (1948) for discrete distributions. Jaynes (1968) showed that (A.2) can be derived by generalizing to continuous distributions, in which case $f(x)^{def}$ appears as an "invariant measure" function (as well as playing the role of default distribution). Once a measure of the amount of uncertainty is established, it then seems reasonable to choose from all distributions f(x) satisfying a set of constraints the one for which S_{CE} is maximal, since this is the distribution that "assumes the least" about the unknown f(x). While agreeing with what is known, it will be maximally noncommittal with regard to all information except the data, and, therefore, the most unbiased estimate.

An independent justification that makes no reference to information theory is given by Shore and Johnson. Jaynes (1957) had already remarked that inferences made using any other information measure than the entropy may lead to contradictions. Shore and Johnson (1980) considered the consequences of requiring that methods of inference be self-consistent. They introduced four axioms that were all based on one fundamental principle: if a problem can be

solved in more than one way, the results should be consistent. Informally, they stated their axioms as follows:

I. *Uniqueness*: The result should be unique,

II *Invariance*: The choice of coordinate system should not matter,

III. *System independence*: It should not matter whether one account for independent information about independent systems separately in terms of different densities or together in terms of a joint density, and

IV. *Subset independence*: It should not matter whether one treats an independent subset of system states in terms of a separate different conditional density or in terms of the full system density.

Shore and Johnson go on to show that given information in the form of a set of constraints on expected values, there is only one distribution satisfying the set of constraints that can be chosen by a procedure that satisfies the four axioms, and this unique distribution can be obtained by maximizing S_{CE}. Therefore, if a method of inference is based on a variational principle, maximizing any function but the cross-entropy will lead to inconsistencies unless that function and entropy have identical maxima (any monotonic function of the entropy will work, for example).

APPENDIX B

LISTING OF THE PROGRAM ISD97

```
        PROGRAM ISD97
C
C       A computer program that analyzes data from a series of in situ
C       measurements on a grid and identifies potential localized areas
C       of elevated activity.
C
        INTEGER M,NX,NY,NXY,MNXY,ISP
        INTEGER DETNUM,PE
        INTEGER MNUM(1000)
        REAL A,HSA
        REAL D(1000),S(1000),X(1000),Y(1000)
        CHARACTER DH*8, EH*8
        COMMON/AISP/A,ISP
        COMMON/MN/M,NX,NY,MNXY,NXY
C
C
        PRINT*,' NAME OF FILE WITH INPUT DATA? (USE 8 CHARACTERS) : '
        READ*,DH
        PRINT*,' NAME OF OUTPUT FILE? (USE 8 CHARACTERS) : '
        READ*,EH
        PRINT*,' AREA OF ELEVATED AREA? (IN SQUARE METERS, e.g.: 1.0) : '
        READ*,HSA
C
C       Next the following data is read in from the file DH:
C       (1) M is the number of measurements.
C       (2) The ground is made up of (NX,NY) cells.
C       (3) A is the size of each cell (in cm).
C       (4) ISP is used by several subroutines.  It is set up
C       normally equal to one half the number of cells between
C       measurements (or slightly more if the number of cells between
C       measurements is not even).  Must have ISP > 0.  ISP is used in
C       the subroutine PEAKXELS to fix the radius of the circle on which
C       the integration over counts is done.
```

```
C       (5) DETNUM is the detector number.
C       (6) PE is an integer, the photon energy.
C       (7) MNUM is the measurement number.  Must Have MNUM(I) > 0.
C       (8) X,Y are the coordinates of measurements (in cm).
C       (9) D are the measurements (in cpm).
C       (10) S are the corresponding sigmas for each measurement.
C
        OPEN (11, FILE=DH, STATUS='OLD')
C
        READ (11,FMT=*) M,NX,NY
        READ (11,FMT=*) A
        READ (11,FMT=*) ISP
        READ (11,FMT=*) DETNUM
        READ (11,FMT=*) PE
        DO 1 I=1,M
          READ (11,FMT=*) MNUM(I),X(I),Y(I),D(I),S(I)
1       CONTINUE
        CLOSE (11)
C
        MNXY = M*NX*NY
        NXY = NX*NY
C
        CALL ISD(DH,EH,HSA,X,Y,D,S,DETNUM,PE,MNUM)
C
        END
C
C       **********************************************************
C
        SUBROUTINE ISD(DH,EH,HSA,X,Y,D,S,DETNUM,PE,MNUM)
C
        INTEGER I,II,K,L,IG
        INTEGER IHP,KL,KU,LL,LU
        INTEGER M,NX,NY,NXY,MNXY,ISP,HSS,G1,G2
        INTEGER DETNUM,PE,NSD,NUM
        INTEGER MNUM(1000)
        REAL C,T,Z,ZAVG,MM,C1,CZ,CME
        REAL HHP,PP,CSQ
        REAL A,HHPNN,HSA
        REAL DAVG,DBKG,SB,XMAX,XMIN,YMAX,YMIN,PPMCF,EAM,SUM1,SUM2
```

```
      REAL  D(M),DSORT(M),EZ(M),E(M),S(M),F(NXY)
      REAL  ERR(M),FP(NXY),PFE(M),RKHS(M),RLHS(M)
      REAL  MHS(M),CHISQ(M)
      REAL  B(MNXY),X(M),Y(M),ZKL(NXY)
      REAL  ZKLP(NXY)
      REAL  FPF(NXY),EF(M)
      REAL  FMAP(NXY)
      REAL  LAMBDA(M),LME(M),BI(M)
      CHARACTER DH*8, EH*8, CH*12, TH*12, WH*12
      CHARACTER UNITS*10
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C     Elevated Areas are modeled as squares of HSS cells per side.
C     HSS is an integer.  Calculate how many cells per elevated area.
C     Calculate new value of HSA from the number of cells per elevated area.
C
      HSS = FLOOR((SQRT(HSA))/(0.01*A))
      HSA = (0.01*A*HSS)**2
      PRINT*,' AREA OF ELEVATED AREA (IN SQUARE METERS) SET TO : ',HSA
      PRINT*,' '
C
      MM = FLOAT(M)
C
C     The user has a choice of two source distributions.
C
      PRINT*,' SOURCE DISTRIBUTION? (UNIFORM=1, PLANE=2) : '
      READ*,NSD
C
C     Calculate the response matrix B.
C
      CALL RESPONSE(B,A,X,Y,DETNUM,PE,UNITS,NSD,PPMCF)
C
C     Set the background activity distribution ZKL to a user chosen
C     value Z (choices 3 or 4 are recommended).  T is the total
C     number of counts.
C
      T = 0.0
      DO 105 I=1,M
```

```fortran
           T = T + D(I)
105    CONTINUE
       DAVG = T/MM
C
       PRINT*,' AVERAGE COUNT RATE = ',DAVG
       PRINT*,' '
C
       SB = 0.0
       DO 107 I=1,MNXY
         SB = SB + B(I)
107    CONTINUE
C
       ZAVG = T/SB
       Z = ZAVG
       DBKG = DAVG
C
       PRINT*,' AVERAGE ACTIVITY (', UNITS, ') = ', ZAVG
       PRINT*,' '
C
108    PRINT*,' COUNT RATE TO USE TO SET A BACKGROUND ACTIVITY? : '
       PRINT*,' 0 - THE AVERAGE VALUE '
       PRINT*,' 1 - THE MEDIAN VALUE '
       PRINT*,' 2 - THE COUNT RATE FROM THE LOWEST MEASUREMENT '
       PRINT*,' 3 - THE WEIGHTED AVERAGE THAT MINIMIZES THE CHI SQUARE '
       PRINT*,' 4 - A USER DETERMINED COUNT RATE :    '
       READ*,IG
       IF (IG .EQ. 1) THEN
         DO 109 I=1,M
           DSORT(I) = D(I)
109      CONTINUE
         CALL HPSORT(M,DSORT)
         K = FLOOR(MM/2.0)
         DBKG = DSORT(K)
         Z = DBKG*(ZAVG/DAVG)
       ELSE IF (IG .EQ. 2) THEN
         DO 110 I=1,M
           DSORT(I) = D(I)
110      CONTINUE
         CALL HPSORT(M,DSORT)
```

```fortran
            DBKG = DSQRT(1)
            Z = DBKG*(ZAVG/DAVG)
          ELSE IF (IG .EQ. 3) THEN
            DO 116 I=1,M
              BI(I) = 0.0
              DO 114 K=1,NX
                DO 112 L=1,NY
                  BI(I) = BI(I) + B(NXY*(I-1)+NY*(K-1)+L)
112             CONTINUE
114           CONTINUE
116         CONTINUE
            SUM1 = 0.0
            SUM2 = 0.0
            DO 118 I=1,M
              SUM1 = SUM1 + D(I)*BI(I)/(S(I)**2)
              SUM2 = SUM2 + (BI(I)/S(I))**2
118         CONTINUE
            Z = SUM1/SUM2
            DBKG = Z*SB/MM
          ELSE IF (IG .EQ. 4) THEN
            PRINT*,' ENTER COUNT RATE (>0) IN cpm : '
            READ*,DBKG
            IF (DBKG .LE. 0.0) THEN
              PRINT*,' COUNT RATE ENTERED MUST BE GREATER THAN ZERO '
              PRINT*,' '
              GO TO 108
            END IF
            Z = DBKG*(ZAVG/DAVG)
          END IF
C
          PRINT*,' BACKGROUND ACTIVITY (', UNITS, ') = ', Z
          PRINT*,' '
C
          DO 211 K=1,NXY
            F(K) = Z
            ZKL(K) = Z
211       CONTINUE
C
C     Calculate the CZ, the chi square of the background distribution.
```

```fortran
C
      C1 = 0.0
      DO 216 I=1,M
        EZ(I) = 0.0
        DO 214 K=1,NX
          DO 212 L=1,NY
            EZ(I) = EZ(I) + B(NXY*(I-1)+NY*(K-1)+L)*F(NY*(K-1)+L)
212       CONTINUE
214     CONTINUE
        C1 = C1+((D(I)-EZ(I))**2)/(S(I)**2)
216   CONTINUE
C
      CZ = C1
C
      PRINT*,' CHI SQUARE/INITIAL GUESS   = ',C1
      PRINT*,' '
C
C     Initialize the lambdas to zero
C
      DO 218 I=1,M
        LAMBDA(I) = 0.0
218   CONTINUE
C
      NUM = 0
      CALL MAXENT(LAMBDA,F,ZKL,D,E,S,B,CME,NUM)
C
C     Save the lambdas from the MaxEnt deconvolution.
C
      DO 220 I=1,M
        LME(I) = LAMBDA(I)
220   CONTINUE
C
C     The FMAP is equal to the background level.
C     FMAP will be used to construct a map with a composite of elevated areas
C     placed on top of a map with background activity Z.
C
      DO 224 K=1,NXY
        FMAP(K) = Z
224   CONTINUE
```

```
C
C   .   The calculation of the locations and strengths of potential elevated
C       areas starts here.  The subroutine PEAKXELS integrates the total
C       excess activity (counts - bkg) surrounding each measurement point and
C       places the result in the array PFE.
C
        CALL PEAKXELS(Z,F,PFE,X,Y)
C
C       For all PFE>0, the optimal position and magnitude for the
C       potential elevated area close to that PFE are determined.
C       FINHS calculates the position, MAGPS or MAGPSNN the magnitude.
C       PP is used to keep track of the intensity of the largest elevated area.
C
        PP = 0.0
C
        DO 305 I=1,M
          MHS(I) = 0.0
          RKHS(I) = 0.0
          RLHS(I) = 0.0
          CHISQ(I) = 0.0
          IF (PFE(I) .GT. 0.0) THEN
            PRINT*,' '
            PRINT*,' CALCULATING PARAMETERS NEAR MEASUREMENT' ,MNUM(I)
            PRINT*,' '
            DO 304 K=1,NXY
              FP(K) = ZKL(K)
              ZKLP(K) = ZKL(K)
304         CONTINUE
C
            CALL FINDHS(X,Y,I,HSS,PFE,RKP,RLP,Z)
            RKHS(I) = RKP
            RLHS(I) = RLP
            CALL MAGPS(D,S,RKP,RLP,I,HHP,CSQ,B,X,Y,ZKL,HSS)
            CALL MAGPSNN(D,S,RKP,RLP,HHPNN,C1,B,A,X,Y,ZKL,HSS)
            IF (HHPNN .GT. HHP) THEN
              HHP = HHPNN
              CSQ = C1
            END IF
            IF (HHP .LT. 1.0) THEN
```

```
                  HHP = 1.0
               END IF
C
C      The elevated area is added to the FP and to the ZKLP (which were
C      set equal to the background activity).  These will be used in a final
C      MaxEnt deconvolution with background + elevated area.
C
               CALL MAKEHS(RKP,RLP,FP,HHP,X,Y,HSS)
               CALL MAKEHS(RKP,RLP,ZKLP,HHP,X,Y,HSS)
C
C      A MaxEnt deconvolution of the FP using as default distribution the ZKLP.
C
C      Initialize the lambdas - set them equal to the LME.
C
               DO 318 III=1,M
                  LAMBDA(III) = LME(III)
318            CONTINUE
C
               NUM = 1
               CALL MAXENT(LAMBDA,FP,ZKLP,D,E,S,B,C,NUM)
C
C      The final magnitude of the elevated area is calculated.
C
               G1 = FLOOR(0.5*HSS)
               G2 = CEILING(0.5*HSS)
               RK = NINT(RKP)
               RL = NINT(RLP)
               KL = RK - G1
               KU = RK + G1
               LL = RL - G1
               LU = RL + G1
               IF (G1 .EQ. G2) THEN
                  RK = FLOOR(RKP)
                  RL = FLOOR(RLP)
                  KL = RK - (G1-1)
                  KU = RK + G1
                  LL = RL - (G1-1)
                  LU = RL + G1
               END IF
```

```fortran
C
            HHP = 0.0
            DO 332 K=KL,KU
              DO 330 L=LL,LU
                HHP = HHP + FP(NY*(K-1)+L)/(Z*HSS*HSS)
330           CONTINUE
332         CONTINUE
C
            CSQ = C
            MHS(I) = HHP
            CHISQ(I) = CSQ
C
C     Print to screen the parameters of this potential ELEVATED AREA.
C
            PRINT*,' '
            PRINT*,' ELEVATED AREA (?)/MEAS. NUMBER = ',MNUM(I)
            PRINT*,' ELEVATED AREA (?)/X COORDINATE = ',RKP
            PRINT*,' ELEVATED AREA (?)/Y COORDINATE = ',RLP
            PRINT*,' ELEVATED AREA (?)/SIZE (x BKG) = ',HHP
            PRINT*,' ELEVATED AREA (?)/CHI SQUARE   = ',CSQ
            PRINT*,' '
C
            CALL MAKEHSMAP(RKP,RLP,FMAP,HHP,X,Y,HSS,Z)
C
C     If this elevated area is the largest one so far, the FP are saved as
C     FPF, and IHP is set to I.  The E are saved as as EF.
C
            IF (MHS(I) .GT. PP) THEN
              IHP = I
              DO 340 II=1,M
                EF(II) = E(II)
340           CONTINUE
              DO 344 K=1,NXY
                FPF(K) = FP(K)
344           CONTINUE
              PP = MHS(I)
            END IF
C
          END IF
```

```fortran
305      CONTINUE
C
C        Print to screen the parameters of the largest potential elevated
C        area found.
C
         RKP = RKHS(IHP)
         RLP = RLHS(IHP)
         HHP = MHS(IHP)
         C1 = CHISQ(IHP)
C
         PRINT*,' '
         PRINT*,' PARAMETERS OF LARGEST POTENTIAL ELEVATED AREA FOUND: '
         PRINT*,' '
         PRINT*,' MEAS. NUMBER = ',MNUM(IHP)
         PRINT*,' X COORDINATE = ',RKP
         PRINT*,' Y COORDINATE = ',RLP
         PRINT*,' SIZE (x BKG) = ',HHP
         PRINT*,' CHI SQUARE   = ',C1
         PRINT*,' '
C
C        Set the FP equal to the FPF.
C
         DO 432 K=1,NXY
            FP(K) = FPF(K)
432      CONTINUE
C
C
C        *** OUTPUT ***
C
         DO 905 I=1,M
            ERR(I) = (EF(I)-D(I))/S(I)
905      CONTINUE
C
         CH = EH//'.MXE'
         TH = EH//'.TBL'
         WH = EH//'.ALL'
C
         OPEN (10, FILE=CH, STATUS='NEW')
         OPEN (12, FILE=EH, STATUS='NEW')
```

```fortran
      OPEN (14, FILE=TH, STATUS='NEW')
      OPEN (16, FILE=WH, STATUS='NEW')
C
C     Set the limits of the activity maps to the rectangle bounded by the
C     maxima and minima of X, Y (in units of cells) plus or
C     minus ISP.
C
      XMAX = X(1)
      XMIN = X(1)
      YMAX = Y(1)
      YMIN = Y(1)
C
      DO 910 I=1,M
        IF (X(I) .GT. XMAX) THEN
          XMAX = X(I)
        END IF
        IF (X(I) .LT. XMIN) THEN
          XMIN = X(I)
        END IF
        IF (Y(I) .GT. YMAX) THEN
          YMAX = Y(I)
        END IF
        IF (Y(I) .LT. YMIN) THEN
          YMIN = Y(I)
        END IF
910   CONTINUE
C
      KU = CEILING(XMAX/A) + ISP
      KL = FLOOR(XMIN/A) - ISP
      LU = CEILING(YMAX/A) + ISP
      LL = FLOOR(YMIN/A) - ISP
C
      IF (KU .GT. NX) THEN
        KU = NX
      END IF
      IF (KL .LT. 1) THEN
        KL = 1
      END IF
      IF (LU .GT. NY) THEN
```

```
            LU = NY
        END IF
        IF (LL .LT. 1) THEN
            LL = 1
        END IF
C
        DO 930 K=KL,KU
            WRITE (10,920)  (F(NY*(K-1)+L),  L = LL,LU)
            WRITE (12,920)  (FP(NY*(K-1)+L),  L = LL,LU)
            WRITE (16,920)  (FMAP(NY*(K-1)+L),  L = LL,LU)
920         FORMAT( 1000 (2X,E8.2) )
930     CONTINUE
C
        WRITE (14,932) EH
932     FORMAT(/2X,'ACTIVITY ARRAY IS STORED IN FILE : ',A)
C
        WRITE (14, 933) DETNUM
933     FORMAT(/2X,'DETECTOR NUMBER : ',I10)
        WRITE (14, 934) PE
934     FORMAT(2X,'PHOTON ENERGY (KeV) = ',I5)
C
        WRITE (14, 935) DAVG
935     FORMAT(/2X,'AVERAGE COUNT RATE (cpm) = ',1P,E10.4)
        WRITE (14, 936) UNITS, ZAVG
936     FORMAT(2X,'AVERAGE ACTIVITY (', A10, ') = ',1P,E10.4)
C
        WRITE (14, 944) DBKG
944     FORMAT(/2X,'AVERAGE BACKGROUND COUNT RATE (cpm) SET TO : ',1P,
       1E10.4)
        WRITE (14, 945) UNITS, Z
945     FORMAT(2X,'BACKGROUND ACTIVITY (', A10, ') SET TO : ',1P,E10.4)
        IF (PPMCF .GT. 0.0) THEN
            WRITE (14, 947) PPMCF
947         FORMAT(2X,'CONVERSION FACTOR (pCi/g TO ppm) USED = ',1P,E10.4,
       1' ppm/(pCi/g)')
        END IF
C
        WRITE (14, 948) CZ
948     FORMAT(/2X,'CHI SQUARE / BACKGROUND ACTIVITY = ',1P,E10.4)
```

```fortran
      WRITE (14, 949) CME
949   FORMAT(2X,'CHI SQUARE / MAXIMUM ENTROPY SOLUTION = ',F10.4)
C
      WRITE (14, 950) IHP
950   FORMAT(/2X,'LARGEST POTENTIAL ELEVATED AREA NEAR MEASUREMENT : '
     1,I3)
      WRITE (14, 951) HSA
951   FORMAT(2X,'AREA  (m^2) = ',F10.4)
      WRITE (14, 952) C1
952   FORMAT(2X,'CHI SQUARE = ',F10.4)
      WRITE (14, 954) HHP
954   FORMAT(2X,'MAGNITUDE (x BACKGROUND) = ',F10.4)
      EAM = HHP*Z
      WRITE (14, 955) UNITS,EAM
955   FORMAT(2X,'MAGNITUDE (', A10, ') = ',1P,E10.4)
C
      WRITE (14, 956)
956   FORMAT(//2X,'NUM/MEAS  CTS/MEAS  CTS/CALC    (E-D)/S=')
      DO 960 I=1,M
        WRITE (14, 958) MNUM(I),D(I),EF(I),ERR(I)
958     FORMAT(2X,I8,2X,1P,E8.2,2X,E8.2,2X,0P,F8.3)
960   CONTINUE
C
      WRITE (14, 970)
970   FORMAT(/2X,'NUM/MEAS    X/HSPT    Y/HSPT    MAG HSPT   CHI SQR')
C
      DO 985 I=1,M
        WRITE (14, 980) I,RKHS(I),RLHS(I),MHS(I),CHISQ(I)
980     FORMAT(2X,I8,2X,F8.2,2X,F8.2,2X,F8.1,2X,F8.2)
985   CONTINUE
C
      CLOSE (10, STATUS='KEEP')
      CLOSE (12, STATUS='KEEP')
      CLOSE (14, STATUS='KEEP')
      CLOSE (16, STATUS='KEEP')
C
      PRINT*,' '
      PRINT*,'  Activity Array is Stored in File : ',EH
C
```

```
9999    RETURN
        END
C
C       ********************************************************
C
        SUBROUTINE PEAKXELS(Z,F,PFE,X,Y)
C
        INTEGER M,NX,NY,NXY,I,K,L,KL,KU,LL,LU,MHD,MHDSQ,RR
        INTEGER ISP
        REAL XX,YY,Z
        REAL A
        REAL F(NXY),PFE(M)
        REAL X(M),Y(M)
        COMMON/AISP/A,ISP
        COMMON/MN/M,NX,NY,MNXY,NXY
C
        DO 20 I=1,M
          MHD = ISP
          MHDSQ = MHD**2
          XX = X(I)/A
          YY = Y(I)/A
          KL = INT(XX) - MHD
          KU = INT(XX) + MHD
          LL = INT(YY) - MHD
          LU = INT(YY) + MHD
C
          IF (KL .LT. 1) THEN
            KL = 1
          END IF
          IF (KU .GT. NX) THEN
            KU = NX
          END IF
          IF (LL .LT. 1) THEN
            LL = 1
          END IF
          IF (LU .GT. NY) THEN
            LU = NY
          END IF
C
```

```
         PFE(I) = 0.0
         DO 10 K=KL,KU
           DO 8 L=LL,LU
             RR = ((INT(XX)-K)**2)+((INT(YY)-L)**2)
             IF (RR .LE. MHDSQ) THEN
               PFE(I) = PFE(I) + F(NY*(K-1)+L) - Z
             END IF
8          CONTINUE
10       CONTINUE
20     CONTINUE
C
       RETURN
       END
C
C      ***********************************************************
C
       SUBROUTINE FINDHS(X,Y,IHP,HSS,PFE,RKP,RLP,Z)
C
C      FINDHS calculates the optimal coordinates (RKP,RLP) of the elevated
C      area.
C
       INTEGER M,NX,NY,NXY,MNXY,ISP,IHP,HSS,IHPK,IHPL,K
       INTEGER I,J,KK,LL,KKL,KKU,LLL,LLU
       REAL DSQ,R2,RR,KO,XH,YH
       REAL A,EPSSQMIN,RKP,RLP,Z
       REAL X(M),XX(M),Y(M),YY(M),PFE(M),PPFE(M)
       REAL RRIJ(M,M),RRHI(M),PHS(NX,NY),EPS(M),EPSSQ(NX,NY)
       COMMON/AISP/A,ISP
       COMMON/MN/M,NX,NY,MNXY,NXY
C
C      R2 is defined in such a way that for a square grid it includes
C      the eight closest measurement points (DSQ is the square of the
C      distance from the center of the square to any of the corners;
C      R2 is the square of {1.5*the distance to closest measurement}).
C      (For a triangular grid it includes the six closest measurements).
C
       DSQ = 2.0*(2.0*A*FLOAT(ISP))**2
       R2 = 1.125*DSQ
C
```

```
C       Calculate (XX,YY), the x and y coordinates for measurement points
C       with PFE>0.  Evaluate K, the number of such measurement points.
C
        K = 0
        DO 10 I=1,M
          RR = (X(IHP)-X(I))**2 + (Y(IHP)-Y(I))**2
          IF (RR .LE. R2) THEN
            IF (PFE(I) .GE. 0.0) THEN
              K = K+1
              XX(K) = X(I)
              YY(K) = Y(I)
              PPFE(K) = PFE(I)
            END IF
          END IF
10      CONTINUE
        KO = FLOAT(K)
C
C       Calculate the square of the distance between detectors that are 100cm
C       above the ground and points on the ground directly under the
C       detectors.  Call it RRIJ(I,J).
C
        DO 14 I=1,K
          DO 12 J=1,K
            RRIJ(I,J) = (XX(I)-XX(J))**2 + (YY(I)-YY(J))**2 + 10000.0
12        CONTINUE
14      CONTINUE
C
        IHPK = NINT(X(IHP)/A)
        IHPL = NINT(Y(IHP)/A)
        KKL = IHPK - 3*ISP
        KKU = IHPK + 3*ISP
        LLL = IHPL - 3*ISP
        LLU = IHPL + 3*ISP
C
        IF (KKL .LT. 1) THEN
          KKL = 1
        END IF
        IF (KKU .GT. NX) THEN
          KKU = NX
```

```
          END IF
          IF (LLL .LT. 1) THEN
            LLL = 1
          END IF
          IF (LLU .GT. NY) THEN
            LLU = NY
          END IF
C
          DO 28 KK=KKL,KKU
            DO 26 LL=LLL,LLU
              XH = A*(FLOAT(KK))
              YH = A*(FLOAT(LL))
              RR = (X(IHP)-XH)**2 + (Y(IHP)-YH)**2
              IF (RR .LE. R2) THEN
C
C     Calculate the optimal activity PHS for a elevated area (point source)
C     at cell coordinates (KK,LL).
C
                PHS(KK,LL) = 0.0
                DO 18 I=1,K
                  RRHI(I) = (XH-XX(I))**2 + (YH-YY(I))**2 + 10000.0
                  DO 16 J=1,K
                    PHS(KK,LL) = PHS(KK,LL)+(RRHI(I)*PPFE(J)/RRIJ(I,J))/K0
16                CONTINUE
18              CONTINUE
C
C     Calculate the additional activity EPS needed at each measurement
C     point to keep the number of counts invariant in the
C     presence of the elevated area.
C
                DO 22 I=1,K
                  EPS(I) = 0.0
                  DO 20 J=1,K
                    EPS(I) = EPS(I) + RRHI(I)*PPFE(I)/RRIJ(I,J)
20                CONTINUE
                  EPS(I) = EPS(I) + PHS(KK,LL)
22              CONTINUE
C
C     Calculate EPSSQ(KK,LL), the sum of the square of the EPSs.
```

```
C
              EPSSQ(KK,LL) = 0.0
              DO 24 I=1,K
                  EPSSQ(KK,LL) = EPSSQ(KK,LL) + (EPS(I))**2
24            CONTINUE
C
           END IF
26     CONTINUE
28   CONTINUE
C
C     Place the elevated area at the cell coordinates (KK,LL) that
C     minimize EPSSQ(KK,LL).  Set (RKP,RLP)=(XH/A,YH/A).
C
       EPSSQMIN = EPSSQ(IHPK,IHPL)
       DO 38 KK=KKL,KKU
         DO 36 LL=LLL,LLU
           XH = A*FLOAT(KK)
           YH = A*FLOAT(LL)
           RR = (X(IHP)-XH)**2 + (Y(IHP)-YH)**2
           IF (RR .LE. R2) THEN
             IF (EPSSQ(KK,LL) .LE. EPSSQMIN) THEN
               EPSSQMIN = EPSSQ(KK,LL)
               RKP = XH/A
               RLP = YH/A
             END IF
           END IF
36       CONTINUE
38     CONTINUE
C
       RETURN
       END
C
C     **********************************************************
C
       SUBROUTINE MAGPS(D,S,RKP,RLP,IHP,HHP,CSQ,B,X,Y,ZKL,HSS)
C
       INTEGER M,NX,NY,NXY,MNXY,I,K,L,KP,LP,IHP
       INTEGER ISP,HSS
       REAL DSQ,RR,R2,RKP,RLP,SUM1,SUM2,HHP,HPS,CSQ
```

```fortran
      REAL  A
      REAL  D(M),S(M),E(M),DE(M),FD(NXY)
      REAL  B(MNXY),X(M),Y(M)
      REAL  ZKL(NXY)
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C     MAGPS scales the point source at (RKP,RLP) to the optimal height,
C     which is given by the multiplicative factor HPS.
C     HPS is calculated by minimizing the chi square FOR THE
C     MEASUREMENTS NEAR THE ELEVATED AREA under the assumption that the
C     surface activity is given by the ZKL with a point source
C     added at (RKP,RLP).
C
C     Set the FD = ZKL
C
      DO 110 K=1,NXY
         FD(K) = ZKL(K)
110   CONTINUE
C
C     Calculate the E due to the background.
C
      DO 6 I=1,M
         E(I) = 0.0
         DO 4 K=1,NX
            DO 2 L=1,NY
               E(I) = E(I) + B(NXY*(I-1)+NY*(K-1)+L)*FD(NY*(K-1)+L)
2           CONTINUE
4        CONTINUE
6     CONTINUE
C
C     Calculate the DE due to activity at the point source.
C
      KP = INT(RKP)
      LP = INT(RLP)
C
      DO 10 I=1,M
         DE(I) = B(NXY*(I-1)+NY*(KP-1)+LP)*FD(NY*(KP-1)+LP)
10    CONTINUE
```

```
C
C       Calculate the optimal activity of the point source.
C
        DSQ = 2.0*(2.0*A*FLOAT(ISP))**2
        R2 = 1.125*DSQ
        SUM1 = 0.0
        SUM2 = 0.0
C
        DO 20 I=1,M
          RR = (X(IHP)-X(I))**2 + (Y(IHP)-Y(I))**2
          IF (RR .LE. R2) THEN
            SUM1 = SUM1 + DE(I)*(D(I)-E(I))/(S(I)**2)
            SUM2 = SUM2 + DE(I)*DE(I)/(S(I)**2)
          END IF
20      CONTINUE
C
        HPS = 1.0 + SUM1/SUM2
C
C       Add the elevated area to the FD.
C
        FD(NY*(KP-1)+LP) = FD(NY*(KP-1)+LP) + (HHP-1.0)*FD(NY*(KP-1)+LP)
C
C       Calculate the chi square for this point source + background.
C
        CSQ = 0.0
        DO 38 I=1,M
          E(I) = 0.0
          DO 37 K=1,NX
            DO 36 L=1,NY
              E(I) = E(I)+B(NXY*(I-1)+NY*(K-1)+L)*FD(NY*(K-1)+L)
36          CONTINUE
37        CONTINUE
          CSQ = CSQ+((D(I)-E(I))**2)/(S(I)**2)
38      CONTINUE
C
C       Set HPP = HPS/(area ELEVATED AREA in cells).
C
        HHP = HPS/(HSS**2)
C
```

```
          RETURN
          END
C
C         **********************************************************
C
          SUBROUTINE MAGPSNN(D,S,RKP,RLP,HHP,CSQ,B,A,X,Y,ZKL,HSS)
C
          INTEGER M,NX,NY,NXY,MNXY,I,K,L,RK,RL,NHS,HSS
          REAL RR,RKP,RLP,HHP,HPS,CSQ,A
          REAL XX,YY,RRN,EBKG,DE
          REAL D(M),S(M),E(M),FD(NXY)
          REAL B(MNXY),X(M),Y(M)
          REAL ZKL(NXY)
          COMMON/MN/M,NX,NY,MNXY,NXY
C
C         HHP is calculated by fitting the point source's magnitude to the
C         total counts at the nearest's measurement point - under the
C         assumption that the surface activity is given by the ZKL
C         with a point source added at (RKP,RLP).
C
C         Set the FD = ZKL
C
          DO 10 K=1,NXY
            FD(K) = ZKL(K)
10        CONTINUE
C
C         Find the measurement point that is nearest to the elevated area.
C
          XX = X(1)/A
          YY = Y(1)/A
          NHS = 1
          RRN = (XX-RKP)**2 + (YY-RLP)**2
          DO 12 I=2,M
            XX = X(I)/A
            YY = Y(I)/A
            RR = (XX-RKP)**2 + (YY-RLP)**2
            IF (RR .LT. RRN) THEN
              NHS = I
              RRN = RR
```

```
         END IF
12       CONTINUE
C
C        Calculate the EBKG for the NHS due to the background.
C
         EBKG = 0.0
         DO 24 K=1,NX
            DO 22 L=1,NY
               EBKG = EBKG + B(NXY*(NHS-1)+NY*(K-1)+L)*FD(NY*(K-1)+L)
22          CONTINUE
24       CONTINUE
C
C        Calculate the DE due to activity at the point source location.
C
         RK = INT(RKP)
         RL = INT(RLP)
C
         DE = DE + B(NXY*(NHS-1)+NY*(RK-1)+RL)*FD(NY*(RK-1)+RL)
C
C        Calculate the (largest) optimal activity of the point source
C        assuming E(NHS) = EBKG+(HPS-1)*DE = D(NHS)+S(NHS).
C
         HPS = 1.0 + (D(NHS)+S(NHS)-EBKG)/DE
C
C        Add the point source to the FD(K,L).
C
         FD(NY*(RK-1)+RL) = FD(NY*(RK-1)+RL) + (HHP-1.0)*FD(NY*(RK-1)+RL)
C
C        Calculate the chi square for this point source + background.
C
         CSQ = 0.0
         DO 38 I=1,M
            E(I) = 0.0
            DO 37 K=1,NX
               DO 36 L=1,NY
                  E(I) = E(I)+B(NXY*(I-1)+NY*(K-1)+L)*FD(NY*(K-1)+L)
36             CONTINUE
37          CONTINUE
            CSQ = CSQ+((D(I)-E(I))**2)/(S(I)**2)
```

```
38      CONTINUE
C
C       Calculate HHP = HPS/(HSS**2).
C
        HHP = HPS/(HSS**2)
C
        RETURN
        END
C
C       ********************************************************
C
        SUBROUTINE MAKEHS(RKP,RLP,FP,HHP,X,Y,HSS)
C
        INTEGER K,L,KL,KU,LL,LU,NX,NY,NXY
        INTEGER ISP,M,HSS,G1,G2
        REAL RKP,RLP,RK,RL,HHP
        REAL A
        REAL FP(NXY)
        REAL X(M),Y(M)
        COMMON/AISP/A,ISP
        COMMON/MN/M,NX,NY,MNXY,NXY
C
C       Model the elevated area with a (HSS**2)-cell area.  MAKEHS scales
C       the elevated area to the optimal height, which is given by the
C       multiplicative factor HHP, and calculates the final FP.
C
C       HHP was calculated in MAGPS or MAGPSNN.
C
        G1 = FLOOR(0.5*HSS)
        G2 = CEILING(0.5*HSS)
        RK = NINT(RKP)
        RL = NINT(RLP)
        KL = RK - G1
        KU = RK + G1
        LL = RL - G1
        LU = RL + G1
        IF (G1 .EQ. G2) THEN
           RK = FLOOR(RKP)
           RL = FLOOR(RLP)
```

```
         KL = RK - (G1-1)
         KU = RK + G1
         LL = RL - (G1-1)
         LU = RL + G1
      END IF
C
      DO 34 K=KL,KU
        DO 32 L=LL,LU
          FP(NY*(K-1)+L) = FP(NY*(K-1)+L) + (HHP-1.0)*FP(NY*(K-1)+L)
32      CONTINUE
34    CONTINUE
C
      RETURN
      END            .
C
C     ************************************************************
C
      SUBROUTINE MAKEHSMAP(RKP,RLP,FP,HHP,X,Y,HSS,Z)
C
      INTEGER K,L,KL,KU,LL,LU,NX,NY,NXY
      INTEGER ISP,M,HSS,G1,G2
      REAL RKP,RLP,RK,RL,HHP
      REAL A,DUMMY,Z
      REAL FP(NXY)
      REAL X(M),Y(M)
      COMMON/AISP/A,ISP
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C     Model the elevated area with a (HSS**2)-cell area.  MAKEHS scales
C     the elevated area to the optimal height, which is given by the
C     multiplicative factor HHP, and calculates the final FP.
C
C     HHP was calculated in MAGPS or MAGPSNN.
C
      G1 = FLOOR(0.5*HSS)
      G2 = CEILING(0.5*HSS)
      RK = NINT(RKP)
      RL = NINT(RLP)
      KL = RK - G1
```

```fortran
          KU = RK + G1
          LL = RL - G1
          LU = RL + G1
          IF (G1 .EQ. G2) THEN
            RK = FLOOR(RKP)
            RL = FLOOR(RLP)
            KL = RK - (G1-1)
            KU = RK + G1
            LL = RL - (G1-1)
            LU = RL + G1
          END IF
C
          DO 34 K=KL,KU
            DO 32 L=LL,LU
              DUMMY = FP(NY*(K-1)+L)/Z
              IF (DUMMY .LT. HHP) THEN
                FP(NY*(K-1)+L) = HHP*Z
              END IF
32          CONTINUE
34        CONTINUE
C
          RETURN
          END
C
C       ***********************************************************
C
          SUBROUTINE hpsort(n,ra)
          INTEGER n
          REAL ra(n)
          INTEGER i,ir,j,l
          REAL rra
          if (n.lt.2) return
          l=n/2+1
          ir=n
10        continue
            if(l.gt.1)then
              l=l-1
              rra=ra(l)
            else
```

```
                  rra=ra(ir)
                  ra(ir)=ra(1)
                  ir=ir-1
                  if(ir.eq.1)then
                    ra(1)=rra
                    return
                  endif
                endif
                i=1
                j=1+1
20              if(j.le.ir)then
                  if(j.lt.ir)then
                    if(ra(j).lt.ra(j+1))j=j+1
                  endif
                  if(rra.lt.ra(j))then
                    ra(i)=ra(j)
                    i=j
                    j=j+j
                  else
                    j=ir+1
                  endif
                goto 20
                endif
                ra(i)=rra
              goto 10
              END
C     (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ************************************************************
C
      SUBROUTINE MAXENT(LAMBDA,F,FDEF,D,E,S,B,C,NUM)
C
      INTEGER M,NX,NY,NXY,MNXY
      INTEGER ITER,NUM
      REAL C,FTOL
      REAL LAMBDA(M),D(M),E(M),S(M)
      REAL F(NXY),FDEF(NXY),B(MNXY),SUM(NX,NY)
      COMMON/MN/M,NX,NY,MNXY,NXY
C
```

```
      FTOL = 1.0E-20
C
      IF (NUM .EQ. 0) THEN
        PRINT*,' STARTING MAXIMUM ENTROPY DECONVOLUTION OF DATA '
        PRINT*,' '
      ELSE IF (NUM .GT. 0) THEN
        PRINT*,' STARTING DECONVOLUTION FOR ACTIVITY DISTRIBUTION '
        PRINT*,' OF BACKGROUND + ELEVATED AREA '
        PRINT*,' '
      END IF
C
      IF (NUM .EQ. 0) THEN
        PRINT*,' STARTING MINIMIZATION OF POTENTIAL FUNCTION '
        PRINT*,' '
        CALL FRPRMN(LAMBDA,M,FTOL,ITER,FRET,FDEF,D,S,B)
        PRINT*,' RESTARTING MINIMIZATION OF POTENTIAL FUNCTION '
        PRINT*,' '
        CALL LMBDAS(FTOL,LAMBDA,ITER,FDEF,D,S,B)
      ELSE IF (NUM .GT. 0) THEN
        CALL LMBDAS(FTOL,LAMBDA,ITER,FDEF,D,S,B)
      END IF
C
      DO 14 K=1,NX
        DO 12 L=1,NY
          SUM(K,L) = 0.0
          DO 10 I=1,M
            SUM(K,L) = SUM(K,L) + LAMBDA(I)*B(NXY*(I-1)+NY*(K-1)+L)
10        CONTINUE
12      CONTINUE
14    CONTINUE
C
      DO 18 K=1,NX
        DO 16 L=1,NY
          F(NY*(K-1)+L) = FDEF(NY*(K-1)+L)*EXP(-SUM(K,L))
16      CONTINUE
18    CONTINUE
C
      C = 0.0
      DO 24 I=1,M
```

```
         E(I) = 0.0
         DO 22 K=1,NX
           DO 20 L=1,NY
             E(I) = E(I) + B(NXY*(I-1)+NY*(K-1)+L)*F(NY*(K-1)+L)
20         CONTINUE
22       CONTINUE
         C = C + ((D(I)-E(I))**2)/(S(I)**2)
24    CONTINUE
C
      IF (NUM .EQ. 0) THEN
        PRINT*,' CHI SQUARE/MAXIMUM ENTROPY = ',C
        PRINT*,' '
      END IF
C
      RETURN
      END
C
C     **************************************************************
C
      FUNCTION FUNC(LAMBDA,FDEF,D,S,B)
C
      INTEGER K,L,M,NX,NY,NXY
      REAL OMEGA,SUM1,SUM2,SUM3,SUM4,SUM5
      REAL LAMBDA(M),FDEF(NXY),D(M),S(M),B(MNXY)
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C     FUNC is the potential function.
C
      OMEGA = FLOAT(M)
C
      SUM1 = 0.0
      SUM5 = 0.0
      DO 14 K=1,NX
        DO 12 L=1,NY
          SUM2 = 0.0
          DO 10 I=1,M
            SUM2 = SUM2 + LAMBDA(I)*B(NXY*(I-1)+NY*(K-1)+L)
10        CONTINUE
          SUM1 = SUM1 + FDEF(NY*(K-1)+L)*EXP(-SUM2)
```

```
            SUM5 = SUM5 + FDEF(NY*(K-1)+L)
12      CONTINUE
14    CONTINUE
C
      SUM3 = 0.0
      SUM4 = 0.0
      DO 16 I=1,M
         SUM3 = SUM3 + (S(I)*LAMBDA(I))**2
         SUM4 = SUM4 + D(I)*LAMBDA(I)
16    CONTINUE
C
      FUNC = -SUM1-SQRT(OMEGA*SUM3)-SUM4+SUM5
      FUNC = -FUNC
C
      RETURN
      END
C
C    ****************************************************************
C
      SUBROUTINE DFUNC(LAMBDA,GRADZ,FDEF,D,S,B)

      INTEGER K,L,M,NX,NY,NXY,DUMMY
      REAL OMEGA,A1,A2,SUM1,SUM2(NX,NY),SUM3
      REAL LAMBDA(M),GRADZ(M),FDEF(NXY),D(M),S(M),B(MNXY)
      COMMON/MN/M,NX,NY,MNXY,NXY
C
C    DFUNC is the gradient of the potential function.
C
      OMEGA = FLOAT(M)
C
      A1 = 0.0
      A2 = 0.0
      DO 8 I=1,M
         A1 = A1 + (S(I)*LAMBDA(I))**2
         A2 = A2 + (S(I))**2
8     CONTINUE

      DO 14 K=1,NX
         DO 12 L=1,NY
```

```fortran
            SUM2(K,L) = 0.0
            DO 10 I=1,M
               SUM2(K,L) = SUM2(K,L) + LAMBDA(I)*B(NXY*(I-1)+NY*(K-1)+L)
10          CONTINUE
12       CONTINUE
14    CONTINUE
C

      DO 30 I=1,M
         SUM1 = 0.0
         DO 18 K=1,NX
            DO 16 L=1,NY
               DUMMY = NXY*(I-1)+NY*(K-1)+L
               SUM1 = SUM1 + FDEF(NY*(K-1)+L)*EXP(-SUM2(K,L))*B(DUMMY)
16          CONTINUE
18       CONTINUE
C

         SUM3 = ((S(I))**2)*(SQRT(OMEGA/A2))
         IF (A1 .GT. 1.0E-20) THEN
            SUM3 = ((S(I))**2)*LAMBDA(I)*(SQRT(OMEGA/A1))
         END IF
C

         GRADZ(I) = SUM1 - SUM3 - D(I)
         GRADZ(I) = -GRADZ(I)
30    CONTINUE
C

      RETURN
      END
C
C     ****************************************************************
C

      SUBROUTINE LMBDAS(FTOL,LAMBDA,ITER,FDEF,D,S,B)

      INTEGER K,L,M,NX,NY,NXY,ITER,ITERMAX
      REAL OMEGA,LB(NX,NY),EPS
      REAL TWOMU,Z,ZOLD,FTOL,P,C
      REAL LAMBDA(M),LAMBDAOLD(M),DL,DLMIN
      REAL FDEF(NXY),D(M),E(M),S(M),B(MNXY)
      COMMON/MN/M,NX,NY,MNXY,NXY

C
```

```fortran
      OMEGA = FLOAT(M)
      ITER = 1
      ITERMAX = 250
      EPS = 1.0E-10
      DLMIN = 0.1
      P = 0.01
      DO 2 I=1,M
        LAMBDAOLD(I) = LAMBDA(I)
2     CONTINUE
      ZOLD = FUNC(LAMBDA,FDEF,D,S,B)
C
      DO 8 K=1,NX
        DO 6 L=1,NY
          LB(K,L) = 0.0
          DO 4 I=1,M
            LB(K,L) = LB(K,L) + LAMBDA(I)*B(NXY*(I-1)+NY*(K-1)+L)
4         CONTINUE
6       CONTINUE
8     CONTINUE
      C = 0.0
      DO 14 I=1,M
        E(I) = 0.0
        DO 12 K=1,NX
          DO 10 L=1,NY
      E(I)=E(I)+FDEF(NY*(K-1)+L)*EXP(-LB(K,L))*B(NXY*(I-1)+NY*(K-1)+L)
10        CONTINUE
12      CONTINUE
        C = C + ((E(I)-D(I))/S(I))**2
14    CONTINUE
C
C     Use the integral equation to modify the lambdas.
C
15    TWOMU = 0.0
      DO 16 I=1,M
        TWOMU = TWOMU + (S(I)*LAMBDA(I))**2
16    CONTINUE
      TWOMU = SQRT(TWOMU/OMEGA)
C
      DO 18 I=1,M
```

```fortran
           LAMBDA(I)  =  (1.0-P)*LAMBDA(I)  +  P*(E(I)-D(I))*TWOMU/(S(I)**2)
18     CONTINUE
       Z = FUNC(LAMBDA,FDEF,D,S,B)
C

       IF (Z .GT. ZOLD) THEN
         DO 24 I=1,M
           LAMBDA(I)  =  LAMBDAOLD(I)
24       CONTINUE
         IF (ABS(C-OMEGA) .LT. SQRT(OMEGA)) THEN
           RETURN
         END IF
         CALL FRPRMN(LAMBDA,M,FTOL,ITER,FRET,FDEF,D,S,B)
         RETURN
       END IF
C

       IF (ITER .GT. ITERMAX) THEN
         DO 26 I=1,M
           LAMBDA(I)  =  LAMBDAOLD(I)
26       CONTINUE
         IF (ABS(C-OMEGA) .LT. SQRT(OMEGA)) THEN
           RETURN
         END IF
         CALL FRPRMN(LAMBDA,M,FTOL,ITER,FRET,FDEF,D,S,B)
         RETURN
       END IF
C

       PRINT*,' SUBROUTINE LMBDAS/NUMBER OF ITERATIONS : ',ITER
       PRINT*,' '
C

       IF (2.0*ABS(Z-ZOLD) .LE. FTOL*(ABS(Z)+ABS(ZOLD)+EPS)) THEN
         RETURN
       END IF
C

       DO 34 K=1,NX
         DO 32 L=1,NY
           LB(K,L)  =  0.0
           DO 30 I=1,M
             LB(K,L)  =  LB(K,L)  +  LAMBDA(I)*B(NXY*(I-1)+NY*(K-1)+L)
30         CONTINUE
```

```fortran
32       CONTINUE
34    CONTINUE
      C = 0.0
      DO 44 I=1,M
        E(I) = 0.0
        DO 42 K=1,NX
          DO 40 L=1,NY
      E(I)=E(I)+B(NXY*(I-1)+NY*(K-1)+L)*FDEF(NY*(K-1)+L)*EXP(-LB(K,L))
40        CONTINUE
42      CONTINUE
        C = C + ((E(I)-D(I))/S(I))**2
44    CONTINUE
C
      IF (P .LT. 0.1) THEN
        DL = 0.0
        DO 46 I=1,M
          DL = DL + (LAMBDA(I)-LAMBDAOLD(I))**2
46      CONTINUE
        DL = SQRT(DL) + EPS
        IF (DL .LT. DLMIN) THEN
          P = P*(DLMIN/DL)
          IF (P .GT. 0.1) THEN
            P = 0.1
          END IF
        END IF
      END IF
C
      DO 48 I=1,M
        LAMBDAOLD(I) = LAMBDA(I)
48    CONTINUE
      ZOLD = Z
      ITER = ITER + 1
      GO TO 15
C
      END
C
C     ****************************************************************
C
      SUBROUTINE frprmn(p,n,ftol,iter,fret,FDEF,D,S,B)
```

- 63 -

```
         INTEGER iter,n,NMAX,ITMAX
         REAL fret,ftol,p(n),EPS,func
         EXTERNAL func
         PARAMETER (NMAX=500,ITMAX=200,EPS=1.e-10)
CU       USES dfunc,func,linmin
         INTEGER its,j
         REAL dgg,fp,gam,gg,g(NMAX),h(NMAX),xi(NMAX)
         REAL FDEF(*),D(*),S(*),B(*)
         fp=func(p,FDEF,D,S,B)
         call dfunc(p,xi,FDEF,D,S,B)
         do 11 j=1,n
           g(j)=-xi(j)
           h(j)=g(j)
           xi(j)=h(j)
11       continue
         do 14 its=1,ITMAX
           iter=its
           PRINT*,' SUBROUTINE FRPRMN/NUMBER OF ITERATIONS : ',ITER
           PRINT*,' '
           call linmin(p,xi,n,fret,FDEF,D,S,B)
           if(2.*abs(fret-fp).le.ftol*(abs(fret)+abs(fp)+EPS))return
           fp=func(p,FDEF,D,S,B)
           call dfunc(p,xi,FDEF,D,S,B)
           gg=0.
           dgg=0.
           do 12 j=1,n
             gg=gg+g(j)**2
C            dgg=dgg+xi(j)**2
             dgg=dgg+(xi(j)+g(j))*xi(j)
12         continue
           if(gg.eq.0.)return
           gam=dgg/gg
           do 13 j=1,n
             g(j)=-xi(j)
             h(j)=g(j)+gam*h(j)
             xi(j)=h(j)
13         continue
14       continue
         pause 'frprmn maximum iterations exceeded'
```

```fortran
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ***************************************************************
C
      SUBROUTINE linmin(p,xi,n,fret,FDEF,D,S,B)
      INTEGER n,NMAX
      REAL fret,p(n),xi(n),TOL
      PARAMETER (NMAX=500,TOL=1.e-6)
CU    USES brent,f1dim,mnbrak
      INTEGER j,ncom
      REAL ax,bx,fa,fb,fx,xmin,xx,pcom(NMAX),xicom(NMAX),brent
      REAL FDEF(*),D(*),S(*),B(*)
      COMMON /f1com/ pcom,xicom,ncom
      EXTERNAL f1dim
      ncom=n
      do 11 j=1,n
        pcom(j)=p(j)
        xicom(j)=xi(j)
11    continue
      ax=0.
      xx=1.
      call mnbrak(ax,xx,bx,fa,fx,fb,f1dim,FDEF,D,S,B)
      fret=brent(ax,xx,bx,f1dim,TOL,xmin,FDEF,D,S,B)
      do 12 j=1,n
        xi(j)=xmin*xi(j)
        p(j)=p(j)+xi(j)
12    continue
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ***************************************************************
C
      SUBROUTINE mnbrak(ax,bx,cx,fa,fb,fc,func,FDEF,D,S,B)
      REAL ax,bx,cx,fa,fb,fc,func,GOLD,GLIMIT,TINY
      EXTERNAL func
      PARAMETER (GOLD=1.618034, GLIMIT=100., TINY=1.e-20)
```

- 65 -

```
          REAL dum,fu,q,r,u,ulim
          REAL FDEF(*),D(*),S(*),B(*)
          fa=func(ax,FDEF,D,S,B)
          fb=func(bx,FDEF,D,S,B)
          if(fb.gt.fa)then
            dum=ax
            ax=bx
            bx=dum
            dum=fb
            fb=fa
            fa=dum
          endif
          cx=bx+GOLD*(bx-ax)
          fc=func(cx,FDEF,D,S,B)
1         if(fb.ge.fc)then
            r=(bx-ax)*(fb-fc)
            q=(bx-cx)*(fb-fa)
            u=bx-((bx-cx)*q-(bx-ax)*r)/(2.*sign(max(abs(q-r),TINY),q-r))
            ulim=bx+GLIMIT*(cx-bx)
            if((bx-u)*(u-cx).gt.0.)then
              fu=func(u,FDEF,D,S,B)
              if(fu.lt.fc)then
                ax=bx
                fa=fb
                bx=u
                fb=fu
                return
              else if(fu.gt.fb)then
                cx=u
                fc=fu
                return
              endif
              u=cx+GOLD*(cx-bx)
              fu=func(u,FDEF,D,S,B)
            else if((cx-u)*(u-ulim).gt.0.)then
              fu=func(u,FDEF,D,S,B)
              if(fu.lt.fc)then
                bx=cx
                cx=u
```

```
                    u=cx+GOLD*(cx-bx)
                    fb=fc
                    fc=fu
                    fu=func(u,FDEF,D,S,B)
                endif
            else if((u-ulim)*(ulim-cx).ge.0.)then
                u=ulim
                fu=func(u,FDEF,D,S,B)
            else
                u=cx+GOLD*(cx-bx)
                fu=func(u,FDEF,D,S,B)
            endif
            ax=bx
            bx=cx
            cx=u
            fa=fb
            fb=fc
            fc=fu
            goto 1
        endif
        return
        END
C   (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C       ****************************************************************
C
        FUNCTION brent(ax,bx,cx,f,tol,xmin,FDEF,DD,S,BB)
        INTEGER ITMAX
        REAL brent,ax,bx,cx,tol,xmin,f,CGOLD,ZEPS
        EXTERNAL f
        PARAMETER (ITMAX=100,CGOLD=.3819660,ZEPS=1.0e-10)
        INTEGER iter
        REAL a,b,d,e,etemp,fu,fv,fw,fx,p,q,r,tol1,tol2,u,v,w,x,xm
        REAL FDEF(*),DD(*),S(*),BB(*)
        a=min(ax,cx)
        b=max(ax,cx)
        v=bx
        w=v
        x=v
```

```
          e=0.
          fx=f(x,FDEF,DD,S,BB)
          fv=fx
          fw=fx
          do 11 iter=1,ITMAX
            xm=0.5*(a+b)
            tol1=tol*abs(x)+ZEPS
            tol2=2.*tol1
            if(abs(x-xm).le.(tol2-.5*(b-a))) goto 3
            if(abs(e).gt.tol1) then
              r=(x-w)*(fx-fv)
              q=(x-v)*(fx-fw)
              p=(x-v)*q-(x-w)*r
              q=2.*(q-r)
              if(q.gt.0.) p=-p
              q=abs(q)
              etemp=e
              e=d
              if(abs(p).ge.abs(.5*q*etemp).or.p.le.q*(a-x).or.p.ge.q*(b-x))
     *goto 1
              d=p/q
              u=x+d
              if(u-a.lt.tol2 .or. b-u.lt.tol2) d=sign(tol1,xm-x)
              goto 2
            endif
1           if(x.ge.xm) then
              e=a-x
            else
              e=b-x
            endif
            d=CGOLD*e
2           if(abs(d).ge.tol1) then
              u=x+d
            else
              u=x+sign(tol1,d)
            endif
            fu=f(u,FDEF,DD,S,BB)
            if(fu.le.fx) then
              if(u.ge.x) then
```

```
                  a=x
              else
                 b=x
              endif
              v=w
              fv=fw
              w=x
              fw=fx
              x=u
              fx=fu
           else
              if(u.lt.x) then
                 a=u
              else
                 b=u
              endif
              if(fu.le.fw .or. w.eq.x) then
                 v=w
                 fv=fw
                 w=u
                 fw=fu
              else if(fu.le.fv .or. v.eq.x .or. v.eq.w) then
                 v=u
                 fv=fu
              endif
           endif
11      continue
        pause 'brent exceed maximum iterations'
3       xmin=x
        brent=fx
        return
        END
C   (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C       ****************************************************************
C
        FUNCTION f1dim(x,FDEF,D,S,B)
        INTEGER NMAX
        REAL f1dim,func,x
```

- 69 -

```
      PARAMETER (NMAX=500)
CU    USES func
      INTEGER j,ncom
      REAL pcom(NMAX),xicom(NMAX),xt(NMAX)
      REAL FDEF(*),D(*),S(*),B(*)
      COMMON /f1com/ pcom,xicom,ncom
      do 11 j=1,ncom
        xt(j)=pcom(j)+x*xicom(j)
11    continue
      f1dim=func(xt,FDEF,D,S,B)
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ****************************************************************
C
      SUBROUTINE RESPONSE(B,A,X,Y,DETNUM,PE,UNITS,NSD,PPMCF)
C
C     Calculates the instrument response ("blur") matrix B.
C     DETNUM is the detector number.
C     PE is an INTEGER, the photon energy in KeV.
C
      INTEGER I,II,K,L
      INTEGER M,NX,NY,NXY,MNXY,IQ
      INTEGER DETNUM,PE,NSD
      REAL AA,RR,XK,XL,R,ASR
      REAL NOP,GPD,TH,ANGC,RHO,AS,PHIOA,ANG2,ANG3
      REAL A,PPMCF
      PARAMETER(J=4)
      INTEGER DET(J)
      REAL B(MNXY),X(M),Y(M)
      CHARACTER UNITS*10
      COMMON/MN/M,NX,NY,MNXY,NXY
C
      DATA DET/75230,2670,30860,1030/
C
      RHO = 1.6
      II = 0
      PPMCF = -1.0
```

```
C
      IF (DETNUM .EQ. DET(1)) THEN
        IF (PE .EQ. 60) THEN
          AA = 2.131E-04
          NOP = 2152.15
          GPD = 0.0383
          AS = 3.968E-01
          ANG2 = 1.625E-02
          ANG3 = -1.513E-04
          II = 1
        ELSE IF (PE .EQ. 583) THEN
          AA = 0.979775E-04
          NOP = 893.60
          GPD = 0.858
          AS = 0.131596
          ANG2 = 0.0
          ANG3 = 0.0
          II = 1
        ELSE IF (PE .EQ. 911) THEN
          AA = 0.799150E-04
          NOP = 680.88
          GPD = 0.29
          AS = 0.106931
          ANG2 = 0.0
          ANG3 = 0.0
          II = 1
        ELSE IF (PE .EQ. 967) THEN
          AA = 0.776824E-04
          NOP = 656.71
          GPD = 0.2295
          AS = 0.1038
          ANG2 = 0.0
          ANG3 = 0.0
          II = 1
        ELSE IF (PE .EQ. 1000) THEN
          AA = 7.645E-05
          NOP = 642.66
          GPD = 0.00845
          AS = 1.021E-01
```

```
      ANG2 = 2.608E-03
      ANG3 = -1.408E-05
      II = 1
    ELSE IF (PE .EQ. 2615) THEN
      AA = 0.463641E-04
      NOP = 358.19
      GPD = 0.998
      AS = 0.0625311
      ANG2 = 0.0
      ANG3 = 0.0
      II = 1
    END IF
  ELSE IF (DETNUM .EQ. DET(2)) THEN
    IF (PE .EQ. 583) THEN
      AA = 0.979775E-04
      NOP = 515.16
      GPD = 0.858
      AS = 0.131596
      ANG2 = 0.0
      ANG3 = 0.0
      II = 1
    ELSE IF (PE .EQ. 609) THEN
      AA = 9.6802E-5
      NOP = 500.42
      GPD = 0.461
      AS = 0.1292
      ANG2 = 0.0
      ANG3 = 0.0
      II = 1
    ELSE IF (PE .EQ. 662) THEN
      AA = 9.1504E-5
      NOP = 473.92
      GPD = 0.852
      AS = 0.12437
      ANG2 = 0.0
      ANG3 = 0.0
      II = 1
    ELSE IF (PE .EQ. 911) THEN
      AA = 0.799150E-04
```

```fortran
         NOP = 383.58
         GPD = 0.29
         AS = 0.106931
         ANG2 = 0.0
         ANG3 = 0.0
         II = 1
       ELSE IF (PE .EQ. 967) THEN
         AA = 0.776824E-04
         NOP = 369.01
         GPD = 0.2295
         AS = 0.1038
         ANG2 = 0.0
         ANG3 = 0.0
         II = 1
       ELSE IF (PE .EQ. 1001) THEN
         AA = 7.645E-05
         NOP = 360.44
         GPD = 0.00845
         AS = 1.021E-01
         ANG2 = 0.0
         ANG3 = 0.0
         II = 1
       ELSE IF (PE .EQ. 2615) THEN
         AA = 0.463641E-04
         NOP = 191.08
         GPD = 0.998
         AS = 0.0625311
         ANG2 = 0.0
         ANG3 = 0.0
         II = 1
       END IF
     ELSE IF (DETNUM .EQ. DET(3)) THEN
       IF (PE .EQ. 583) THEN
         AA = 0.979775E-04
         NOP = 836.42
         GPD = 0.858
         AS = 0.131596
         ANG2 = 0.0
         ANG3 = 0.0
```

```
      II = 1
    ELSE IF (PE .EQ. 609) THEN
      AA = 9.6802E-5
      NOP = 811.86
      GPD = 0.461
      AS = 0.1292
      ANG2 = 0.0
      ANG3 = 0.0
      II = 1
    ELSE IF (PE .EQ. 911) THEN
      AA = 0.799150E-04
      NOP = 620.93
      GPD = 0.29
      AS = 0.106931
      ANG2 = 0.0
      ANG3 = 0.0
      II = 1
    ELSE IF (PE .EQ. 967) THEN
      AA = 0.776824E-04
      NOP = 597.57
      GPD = 0.2295
      AS = 0.1038
      ANG2 = 0.0
      ANG3 = 0.0
      II = 1
    ELSE IF (PE .EQ. 1001) THEN
      AA = 7.645E-05
      NOP = 584.57
      GPD = 0.00845
      AS = 1.021E-01
      ANG2 = 0.0
      ANG3 = 0.0
      II = 1
    ELSE IF (PE .EQ. 2615) THEN
      AA = 0.463641E-04
      NOP = 312.17
      GPD = 0.998
      AS = 0.0625311
      ANG2 = 0.0
```

```
         ANG3 = 0.0
         II = 1
       END IF
     ELSE IF (DETNUM .EQ. DET(4)) THEN
       IF (PE .EQ. 662) THEN
         AA = 9.1504E-5
         NOP = 234.35
         GPD = 0.852
         AS = 0.12437
         ANG2 = 0.0
         ANG3 = 0.0
         II = 1
       END IF
     ELSE
       PRINT*,' '
       PRINT*,'  THE NUMBER OF THE DETECTOR DOES NOT MATCH ANY OF THE '
       PRINT*,'  DETECTOR NUMBERS LISTED IN THE SUBROUTINE RESPONSE. '
       PRINT*,' '
       PRINT*,'  PLEASE CHECK YOUR INPUT FILE. '
       STOP
     END IF
C
     IF (II .EQ. 0) THEN
       PRINT*,' '
       PRINT*,'  THE PHOTON ENERGY DOES NOT MATCH ANY OF THE ENERGIES'
       PRINT*,'  LISTED FOR THIS DETECTOR IN THE SUBROUTINE RESPONSE. '
       PRINT*,' '
       PRINT*,'  PLEASE CHECK YOUR INPUT FILE. '
       STOP
     END IF
C
     IF (NSD .EQ. 1) THEN
       PRINT*,' CALCULATING RESPONSE MATRIX / UNIFORM DISTRIBUTION'
       PRINT*,' '
       PRINT*,' ENTER YOUR CHOICE OF UNITS FOR THE SPECIFIC ACTIVITY,'
       PRINT*,' 0 FOR Bq/g      '
       PRINT*,' 1 FOR pCi/g     '
       PRINT*,' 2 FOR ppm    : '
       READ*,IQ
```

```fortran
          UNITS = ' Bq/g'
          IF (IQ .EQ. 1) THEN
            UNITS = 'pCi/g'
            NOP = NOP/27.03
          ELSE IF (IQ .EQ. 2) THEN
            PRINT*,' ENTER CONVERSION FACTOR (FROM pCi/g TO ppm) FOR THE'
            PRINT*,' RADIONUCLIDE BEING MEASURED: '
            READ*,PPMCF
            UNITS = '  ppm'
            NOP = (NOP/27.03)/PPMCF
          END IF
C
          DO 15 I=1,M
            DO 10 K=1,NX
              DO 5 L=1,NY
                XK = A*(FLOAT(K))
                XL = A*(FLOAT(L))
                RR = ((X(I)-XK)**2)+((Y(I)-XL)**2)+10000.0
                R = SQRT(RR)
                ASR = AS*R
                TH = 57.30*ACOS(100.0/R)
                ANGC = 1.0 + ANG2*TH + ANG3*(TH**2)
                PHIOA = (A**2)*RHO*(100.0/R)/12.566
                IF (ASR .GT. 10.0) THEN
                  PHIOA = PHIOA*(1.0/R)*(EXPZE2Z(ASR))
                ELSE
                  PHIOA = PHIOA*(EXP(ASR)/R)*(EXPINT(2,ASR))
                END IF
                B(NXY*(I-1)+NY*(K-1)+L) = EXP(-AA*R)*PHIOA*NOP*GPD*ANGC
5             CONTINUE
10          CONTINUE
15        CONTINUE
        ELSE IF (NSD .EQ. 2) THEN
          PRINT*,' CALCULATING RESPONSE MATRIX - PLANE DISTRIBUTION'
          PRINT*,' '
          PRINT*,' ENTER YOUR CHOICE OF UNITS FOR THE SPECIFIC ACTIVITY,'
          PRINT*,' O FOR Bq/m**2    '
          PRINT*,' 1 FOR pCi/m**2 :'
          READ*,IQ
```

```
          UNITS = ' Bq/m**2'
          IF (IQ .EQ. 1) THEN
            UNITS = 'pCi/m**2'
            NOP = NOP/27.03
          END IF
C
          DO 35 I=1,M
            DO 30 K=1,NX
              DO 25 L=1,NY
                XK = A*(FLOAT(K))
                XL = A*(FLOAT(L))
                RR = ((X(I)-XK)**2)+((Y(I)-XL)**2)+10000.0
                R = SQRT(RR)
                TH = 57.30*ACOS(100.0/R)
                ANGC = 1.0 + ANG2*TH + ANG3*(TH**2)
                PHIOA = ((A**2)/10000.0)/(12.566*RR)
                B(NXY*(I-1)+NY*(K-1)+L) = EXP(-AA*R)*PHIOA*NOP*GPD*ANGC
25            CONTINUE
30          CONTINUE
35        CONTINUE
        ELSE
          PRINT*,' '
          PRINT*,'  THE SOURCE DISTRIBUTION ENTERED DOES NOT MATCH ANY'
          PRINT*,'  OF THE DISTRIBUTIONS LISTED FOR THIS DETECTOR'
          PRINT*,'  IN THE SUBROUTINE RESPONSE. '
          PRINT*,' '
          STOP
        END IF
C
        RETURN
        END
C
C        ****************************************************************
C
        FUNCTION expint(n,x)
        INTEGER n,MAXIT
        REAL expint,x,EPS,FPMIN,EULER
        PARAMETER (MAXIT=100,EPS=1.e-7,FPMIN=1.e-30,EULER=.5772156649)
        INTEGER i,ii,nm1
```

- 77 -

```
      REAL a,b,c,d,del,fact,h,psi
      nm1=n-1
      if(n.lt.0.or.x.lt.0..or.(x.eq.0..and.(n.eq.0.or.n.eq.1)))then
        pause 'bad arguments in expint'
      else if(n.eq.0)then
        expint=exp(-x)/x
      else if(x.eq.0.)then
        expint=1./nm1
      else if(x.gt.1.)then
        b=x+n
        c=1./FPMIN
        d=1./b
        h=d
        do 11 i=1,MAXIT
          a=-i*(nm1+i)
          b=b+2.
          d=1./(a*d+b)
          c=b+a/c
          del=c*d
          h=h*del
          if(abs(del-1.).lt.EPS)then
            expint=h*exp(-x)
            return
          endif
11      continue
        pause 'continued fraction failed in expint'
      else
        if(nm1.ne.0)then
          expint=1./nm1
        else
          expint=-log(x)-EULER
        endif
        fact=1.
        do 13 i=1,MAXIT
          fact=-fact*x/i
          if(i.ne.nm1)then
            del=-fact/(i-nm1)
          else
            psi=-EULER
```

```
          do 12 ii=1,nm1
            psi=psi+1./ii
12          continue
          del=fact*(-log(x)+psi)
        endif
        expint=expint+del
        if(abs(del).lt.abs(expint)*EPS) return
13      continue
        pause 'series failed in expint'
      endif
      return
      END
C   (C) Copr. 1986-92 Numerical Recipes Software %-,.
C
C     ********************************************************************
C
      FUNCTION EXPZE2Z(X)
C
C     Function to calculate exp(x)*E2(x), where E2(x) is an exponential
C     integral.  Uses the rational approximation by Paul Verbeeck,
C     "Rational approximations for exponential integrals En(x)", in
C     Bulletin de l'Academie royale de Belgique (Classe de Sciences),
C     page 1064, 5e Serie - Tome LVI, 1970.
C
C     exp(x)*E2(x) = (1/x)*(v1(z)/v2(z)) where v1,v2 are third order
C     polynomials, z=1/x, x>1, and stated maximum error < 9.E-06.
C
      REAL X,A2,A3,B2,B3
C
      A2 = 2.6731571
      A3 = 0.1212641
      B2 = 4.6660820
      B3 = 3.7340708
C
      EXPZE2Z = (1.0+(A2/X)+(A3/(X**2)))/(X*(1.0+(B2/X)+(B3/(X**2))))
C
      RETURN
      END
C
C     ********************************************************************
```

INTERNET DOCUMENT INFORMATION FORM

A . Report Title: ISD97, A Computer To Analyze Data From A Series of In Situ Measurements on a Grid and Identify Potential Localized Areas of Elevated Activity

B. DATE Report Downloaded From the Internet   8/31/98

C. Report's Point of Contact: (Name, Organization, Address, Office Symbol, & Ph #):   Environmental Measurements Lab U.S. Department of Energy 201 Varick Street, 5th Floor New York, NY 10014-4811

D. Currently Applicable Classification Level: Unclassified

E. Distribution Statement A: Approved for Public Release

F.` The foregoing information was compiled and provided by: DTIC-OCA, Initials: _UM_ _____**Preparation Date:** 8/31/98